UNIVERSITY OF ALBERTA
LIBRARY

Regulations Regarding Theses and Dissertations

Typescript copies of theses and dissertations for Master's and Doctor's degrees deposited in the University of Alberta Library, as the official Copy of the Faculty of Graduate Studies, may be consulted in the Reference Reading Room only.

A second copy is on deposit in the Department under whose supervision the work was done. Some Departments are willing to loan their copy to libraries, through the inter-library loan service of the University of Alberta Library.

These theses and dissertations are to be used only with due regard to the rights of the author. Written permission of the author and of the Department must be obtained through the University of Alberta Library when extended passages are copied. When permission has been granted, acknowledgement must appear in the published work.

This thesis or dissertation has been used in accordance with the above regulations by the persons listed below. The borrowing library is obligated to secure the signature of each user.

THE UNIVERSITY OF ALBERTA

ERROR ANALYSES FOR CERTAIN MATRIX EQUATIONS

by

Vinod K. Arora

A THESIS

SUBMITTED TO THE FACULTY OF GRADUATE STUDIES

IN PARTIAL FULFILMENT OF THE REQUIREMENTS FOR THE DEGREE

OF MASTER OF SCIENCE

DEPARTMENT OF COMPUTING SCIENCE

EDMONTON, ALBERTA

July, 1968

UNIVERSITY OF ALBERTA

FACULTY OF GRADUATE STUDIES

The undersigned certify that they have read, and recommend to the Faculty of Graduate Studies for acceptance, a thesis entitled ERROR ANALYSES FOR CERTAIN MATRIX EQUATIONS submitted by Vinod K. Arora in partial fulfilment of the requirements for the degree of Master of Science.

ABSTRACT

This thesis reviews various algorithms for computing solutions of systems of linear algebraic equations and for computing the inverse of general matrices, block-symmetric matrices, and band matrices. Norm bounds for the round-off error incurred in performing these computations are obtained. Inverses of several matrices and solutions of several systems of equations, are computed numerically, and a comparison is made between the predicted bounds of error and the actual round-off errors.

ACKNOWLEDGEMENTS

I wish to express my sincere appreciation to Professor S. Charmonman for his invaluable supervision and guidance in the preparation of this thesis, and to Professor D.B. Scott, Head of the Department of Computing Science, University of Alberta, for providing necessary facilities and financial assistance while this research was being done. I also wish to thank National Research Council of Canada for financial assistance provided through the operating grant No. NRC A-4076.

TABLE OF CONTENTS

LIST OF TABLES

CHAPTER I

INTRODUCTION

1.1 History

The development of automatic digital computers has tremendously increased the capability of a numerical analyst to carry out numerical solution of a mathematical problem involving a very large number of arithmetic operations. But, due to the amount of computation involved, the intermediate results are so numerous so as to preclude careful scrutiny by the analyst. Consequently, it could happen that due to the accumulation of round-off errors, the final results obtained may be completely meaningless in terms of the original problem. Therefore, if the speed of an electronic digital computer has relegated the simplicity of the numerical computation to a somewhat secondary position, it has also established a need for studying the cumulative effect of round-off errors made in performing numerical computations.

The problem of estimation of round-off errors in the solution of linear algebraic equations and inversion of matrices was first considered by Von Neumann and Goldstine [13], and Turing [12]. The former gave rigorous

error analyses of the techniques based on Gaussian elimination and may be said to have laid the foundation of modern error analysis. An outstanding contribution to the field of error analysis has been made by Wilkinson [14-19] whose researches on this topic have been the subject matter of several papers and books.

1.2 Purpose of Study

An important criterion for the selection of a particular method for the solution of a system of linear algebraic equations and inversion of matrices is that the method which leads to smaller error bounds for the round-off errors incurred in performing numerical computations is better, other considerations being equal. However, the actual errors depend on the nature of elements of the matrices considered. In this paper, we attempt to find upper bounds for the round-off errors incurred in the computation of the solution of a system of linear algebraic equations and inversion of (a) general matrices using Schur's identity [9]; (b) block-symmetric matrices using computational procedures due to Charmonman [1]; (c) band matrices using minimum storage.

CHAPTER II

BASIC THEORY

2.1 Matrix Algebra

We refer in this section, to some conventions of notations and discuss some of the properties of matrices which will be required repeatedly in the remainder of this paper. Some important definitions including the concept of matrix norm of square and rectangular matrices are also introduced.

2.1.1 Basic Notations and Definitions

We denote matrices by upper-case Roman and Greek letters, and vectors and scalars by lower-case Roman letters. The (i,j) element of a matrix A is denoted by a_{ij}. We say that a rectangular matrix $A = (a_{ij})$ consisting of m rows and n columns is of dimension m-by-n. If the number of rows and columns of the matrix are equal, the matrix A is square and of order n. In general, matrices will be square and of order n, and vectors of dimension n, unless it is indicated otherwise. The identity matrix is denoted by I and the null matrix by 0. The notation $|A|$ is used for the matrix which has $|a_{ij}|$, the absolute value of a_{ij} for its (i,j) element and the

notation det (A) for the determinant of a square matrix A. The inverse of a non-singular square matrix A is denoted by A^{-1}. If the matrix A is real, then A^T denotes the transpose of A. The matrix A is called symmetric when $A = A^T$. A matrix A of order $2n$ is called block-symmetric when it can be partitioned as $\left[\begin{array}{c|c} C & D \\ \hline D & C \end{array}\right]$, where C and D are square matrices of order n. A matrix A is said to be a band matrix of band-width $(2p+1)$ if $a_{ij} = 0$ for all i and j such that $|i-j|>p$. We denote by L a lower-triangular matrix in which all elements above the main diagonal are zero. The lower-triangular matrix L is called unit lower-triangular if the elements on the main diagonal are all unity. Similarly, we denote by U an upper-triangular matrix in which all elements below the main diagonal are zero, and if the main diagonal elements are all unity, the matrix U is called unit upper-triangular.

2.1.2 Norms of Vectors A norm of a vector x is a non-negative scalar $\|x\|$ which is a measure of the size of the vector, and satisfies the conditions

$$\text{(i)} \quad \|x\| > 0 \quad \text{for} \quad x \neq 0 \quad \text{and} \quad \|0\| = 0;$$

$$(2.1) \qquad \text{(ii)} \quad \|kx\| = |k|\,\|x\| \quad \text{for any scalar } k;$$

$$\text{(iii)} \quad \|x+y\| \leq \|x\| + \|y\| .$$

Three different norms of vectors which satisfy the above conditions are:

$$(2.2) \qquad \|x\|_1 = \sum_{i=1}^{n} |x_i| ,$$

$$(2.3) \qquad \|x\|_\infty = \max_i |x_i| ,$$

and

$$(2.4) \qquad \|x\|_E = \left(\sum_{i=1}^{n} |x_i|^2 \right)^{\frac{1}{2}} .$$

2.1.3 <u>Norms of Matrices</u> A norm of a square matrix A is a non-negative scalar $\|A\|$ which is a measure of the magnitude of the matrix, and satisfies the conditions

$$\text{(i)} \quad \|A\| > 0 \quad \text{if} \quad A \neq 0 \quad \text{and} \quad \|0\| = 0 ;$$

(2.5) (ii) $\|kA\| = |k| \; \|A\|$ for any scalar k ;

(iii) $\|A+B\| \leq \|A\|+\|B\|$;

(iv) $\|AB\| \leq \|A\| \; \|B\|$.

The following matrix norms are in common use, each of which satisfies the above conditions:

(2.6) $$\|A\|_1 = \max_j \sum_{i=1}^{n} |a_{ij}| \, ,$$

(2.7) $$\|A\|_\infty = \max_i \sum_{j=1}^{n} |a_{ij}| \, ,$$

(2.8) $$\|A\|_2 = (\text{maximum eigenvalue of } A^T A)^{\frac{1}{2}} \, ,$$

and

(2.9) $$\|A\|_E = \left(\sum_{i=1}^{n} \sum_{j=1}^{n} |a_{ij}|^2\right)^{\frac{1}{2}} .$$

The definition of matrix norm has been extended by Householder [6], and Korganoff and Pavel-Parvu [8] to include rectangular matrices. A rectangular matrix can be normed by adjoining null rows and columns so as

to make it a square matrix. Once this is done, the definition of norm of a square matrix may be applied.

2.1.4 Some Elementary Properties of Norms In this section we assemble certain inequalities involving vector and matrix norms which we will use repeatedly. Throughout this paper, we will only use the row norm or ∞-norm, since it can be easily computed. The following properties hold for norms defined in Subsection 2.1.3, unless it is indicated otherwise. For the proof of these properties see for example, [11] or [16].

(A) $$\|x-y\| \geq \begin{cases} \|x\|-\|y\| , \\ \|y\|-\|x\| . \end{cases}$$

(B) $$\|A-B\| \geq \begin{cases} \|A\|-\|B\| , \\ \|B\|-\|A\| . \end{cases}$$

(C) $$\| \, |x| \, \| = \|x\| ,$$

and

$\| \, |A| \, \| = \|A\|$, for any of our norms except the spectral norm, (2.8).

(D) If $\|A\| < 1$, then $(I+A)$ is non-singular.

(E) If $\|A\| < 1$, then

(i) $$\|(I+A)^{-1}\| \leq \frac{1}{1-\|A\|} ,$$

and

(ii) $$\|(I+A)^{-1} - I\| \leq \frac{\|A\|}{1-\|A\|} .$$

(F) If A is non-singular and $\|A^{-1}\| \; \|E\| < 1$, then $(A+E)$ is non-singular and hence

(i) $$\|(A+E)^{-1} - A^{-1}\| \leq \frac{\|A^{-1}\| \|A^{-1}\| \|E\|}{1-\|A^{-1}\| \|E\|} ,$$

and

(ii) $$\|(A+E)^{-1}\| \leq \frac{\|A^{-1}\|}{1-\|A^{-1}\| \|E\|} .$$

2.2 Gaussian Elimination

In this section we describe one of the best known and most widely used 'direct method' for solving linear systems of algebraic equations and for inverting matrices,

due to Gauss [5]. A direct method is one in which after a finite number of operations a computed result is obtained which would be the same as the true solution of the given system if the computations are carried without round-off. Basically, the Gaussian elimination method is simple. The first equation in the given system is used to eliminate the first unknown from the last $(n-1)$ equations, then the new second equation is used to eliminate the second unknown from the last $(n-2)$ equations, etc. The process of forward elimination is completed in $(n-1)$ reductions. The resulting system which is equivalent to the original system is upper-triangular and can be solved by back-substitution.

Let

$$(2.10)\quad A=A^{(1)} = \begin{bmatrix} a_{11}^{(1)} & a_{12}^{(1)} & \cdots & a_{1n}^{(1)} \\ a_{21}^{(1)} & a_{22}^{(1)} & \cdots & a_{2n}^{(1)} \\ \vdots & \vdots & & \vdots \\ a_{n1}^{(1)} & a_{n2}^{(1)} & \cdots & a_{nn}^{(1)} \end{bmatrix},$$

be the coefficient matrix of a system of linear equations of order n and rank n. Then after k reductions, the

reduced matrix $A^{(k)}$ becomes

$$(2.11)\quad A^{(k)} = \begin{bmatrix} a_{11}^{(1)} & a_{12}^{(1)} & \cdots & a_{1k-1}^{(1)} & a_{1k}^{(1)} & \cdots & a_{1n}^{(1)} \\ 0 & a_{22}^{(2)} & \cdots & a_{2k-1}^{(2)} & a_{2k}^{(2)} & \cdots & a_{2n}^{(2)} \\ \vdots & 0 & & \vdots & \vdots & & \vdots \\ 0 & 0 & \cdots & a_{k-1,k-1}^{(k-1)} & a_{k-1,k}^{(k-1)} & \cdots & a_{k-1,n}^{(k-1)} \\ 0 & 0 & & 0 & \vdots & & \vdots \\ \vdots & \vdots & & \vdots & \vdots & & \vdots \\ 0 & 0 & \cdots & 0 & a_{nk}^{(k)} & \cdots & a_{nn}^{(k)} \end{bmatrix}$$

where

$$(2.12)\quad a_{ij}^{(k)} = a_{ij}^{(k-1)} - \frac{a_{ik}^{(k-1)}}{a_{kk}^{(k-1)}} a_{kj}^{(k-1)} \qquad \begin{array}{l} k=2,\ldots,n\ , \\ j=k,\ldots,n\ , \\ i=k,\ldots,n\ . \end{array}$$

The solution $x = (x_1, x_2, \ldots, x_n)$ to the system of equations $Ax = b$ is easily computed from the n^{th} reduced matrix $A^{(n)}$ by

$$(2.13)\quad x_i = \frac{1}{a_{ii}^{(i)}} [a_{i,n+1}^{(i)} - \sum_{j=i+1}^{n} a_{ij}^{(i)} x_j]\ ,\ i=n,\ldots,1,$$

where $a_{i,n+1}^{(i)}$, $i=1,2,\ldots,n$ denote the components of

the right-hand vector in the n^{th} reduced system.

The algebraic basis of the Gaussian elimination is contained in the following theorem:

Theorem 2.1 *Given a square matrix* A *of order* n, *let* A_k *denote the principal minor matrix constructed from first* k *rows and columns. Assume that* $\det(A_k) \neq 0$ *for* $k=1,2,\ldots,n-1$. *Then there exists a unique unit lower-triangular matrix* $L=(m_{ij})$ *and a unique upper-triangular matrix* $U=(u_{ij})$ *so that* $LU=A$. *Moreover,* $\det(A) = \prod_{i=1}^{n} u_{ii}$, *where* $\det(A)$ *denotes the determinant of matrix* A.

A proof of this theorem is given in [4].

2.2.1 A Need for Pivoting In using Gaussian elimination with natural ordering we have ignored the possibility of the break-down of the process due to one of the diagonal elements having the value zero. Ignoring the possibility of vanishing of the last diagonal element in the n^{th} reduced matrix $A^{(n)}$, in which case the matrix is singular, let us consider that an intermediate diagonal term becomes zero. It means only that a leading submatrix

of A has a zero determinant, and does not necessarily imply the singularity of A. Let us assume that $a_{11}=0$ and since $\det(A)\neq 0$, we can find an $a_{i1}\neq 0$ for some $i>1$. Thus zero pivotal element can be avoided by interchange of the i^{th} row with the first row. This procedure can be employed at all stages of reduction. In theory, interchange is necessary only when a pivotal element is exactly zero. However, in practice, this procedure is satisfactory only when the magnitude of the elements of A and the successive derived systems do not vary appreciably in size. But, when any of the pivotal elements $a_{ii}^{(i)}$ is close to zero or small in magnitude compared to $a_{ki}^{(i)}$, $k>i$, then a serious round-off error may be incurred since the percentage error in the reciprocal of a number varies inversely with the magnitude of the number. To avoid this we may use pivoting strategies, either partial or complete, [18].

2.2.1.1 <u>Partial Pivoting</u> At the i^{th} stage in the reduction of A into $A^{(k)}$, we locate the element of maximum magnitude in rows i through n. Suppose the pivotal element is in the j^{th} row, then rows i and j are interchanged. The interchange of rows can be

accomplished by premultiplying the i^{th} reduced matrix by the permutation matrix R_{ij}. However, in practice physical interchange of rows can be avoided by constructing a vector of integers $p_i (i=1,2,..,n)$ and calling for $a_{p_i,j}$ instead of a_{ij}.

2.2.1.2 Complete Pivoting By complete pivoting we mean the following: at the i^{th} stage in the reduction of A into $A^{(k)}$, we locate the element of the maximum magnitude in rows i through n and columns i through n, i.e., we find the element of maximum modulus in the matrix formed by deleting the first $i-1$ rows and columns of $A^{(i)}$. The pivotal element is transferred to position (i,i) of matrix $A^{(i)}$ by one row interchange and one column interchange. This interchange can be effected again by use of permutation matrices.

2.3 Error Analysis

In this section we give an account of the error analysis of some of the fundamental mathematical computations performed on a digital computer using floating-point arithmetic. We consider only floating-point arithmetic since it is available in all modern digital

computers. Moreover, fixed-point arithmetic is not suitable for handling numbers of large and varying magnitudes in scientific computation. All the results presented in this section are due to Wilkinson [17].

There are two main forms of error analyses that can be used. We may attempt to trace the forward propogation of individual rounding errors and then compare the computed numbers with those which exact computation would produce or we may show that computed results may be obtained by performing the exact computation on a perturbed problem. The former is referred to as 'forward analysis' and the latter 'backward analysis'. We will only consider the backward error analysis since this is often much simpler, particularly if floating-point arithmetic is used. Assume that we calculate a quantity x given by the mathematical expression

$$x = f(x_1,x_2,\ldots,x_n) \; . \tag{2.14}$$

It follows using backward analysis that the computed x satisfies exactly an equation of the form

$$x = f(x_1+\varepsilon_1,x_2+\varepsilon_2,\ldots,x_n+\varepsilon_n) \; , \tag{2.15}$$

where ε_i are the perturbations. We attempt to place bounds on the magnitude of the perturbations ε_i. Corresponding to the mathematical equation (2.14) defining x, we have a computational equation which defines the computed value of x, as in (2.15) followed by bounds for $\varepsilon_i (i=1,2,..,n)$.

In binary floating-point arithmetic each number x is represented in the form $x=2^b(a)$, where b is a positive or negative integer, and a is a number which satisfies $\frac{1}{2} \leq |a| \leq 1$. The number a is called the mantissa or fractional part and b is called the exponent or index of x. We will denote by t, the number of binary digits in the mantissa of a floating-point number.

2.3.1 <u>Common Floating-Point Operations</u> Let the number ε denote the unit round-off error, i.e.,

$$(2.16) \qquad |\varepsilon| \leq 2^{-t} ,$$

and $f\ell$ denote a floating-point arithmetic computation. Then the following theorem gives a bound on the round-off errors incurred in performing basic arithmetic

operations in floating-point using a double-precision accumulator:

Theorem 2.2 *Given two floating-point numbers* $x_1 = 2^{b_1}(a_1)$ *and* $x_2 = 2^{b_2}(a_2)$,

$$(2.17) \qquad fl(x_1 * x_2) \equiv (x_1 * x_2)(1+\varepsilon) ,$$

where $*$ *denote any of the operators* $+,-,\times,\div$ *and* ε *is defined by* (2.16).

For the proof of this theorem see [17,pp.7-9] or [4,pp.88-89].

2.3.2 Inner Product Computations Throughout this paper we assume that floating-point computations are performed with a precision of t binary digits on a computer with a double-precision accumulator but without accumulation of inner products. The inner product $\sum_{i=1}^{n} a_i b_i$ without accumulation in floating-point is computed as follows: the products $a_i b_i$ are computed with $2t$ digits and rounded to t digits. The resulting products are then added in the order in which they are written using

a double-precision accumulator.

Theorem 2.3 *Using a double-precision accumulator without floating-point accumulation,*

$$(2.18) \qquad f\ell\left(\sum_{i=1}^{n} a_i b_i\right) \equiv \sum_{i=1}^{n} a_i b_i (1+\varepsilon_i) ,$$

where

$$(2.19) \qquad |\varepsilon_1| \leq n 2^{-t_1} ,$$

$$(2.20) \qquad |\varepsilon_r| \leq (n-r+2) 2^{-t_1}, \qquad r=2,3,\ldots,n,$$

and

$$(2.21) \qquad 2^{-t_1} = 1.06\ 2^{-t} ,$$

provided

$$(2.22) \qquad n 2^{-t_1} < 0.1 .$$

A proof of this theorem is given in [17,pp.18-19].

2.3.3 <u>Matrix Operations</u> We will repeatedly require the round-off error bounds for the matrix addition and multiplication. Theorems 2.2 and 2.3 can be used to prove the following two theorems:

<u>Theorem 2.4</u> *If A and B are matrices of the same dimensions, then using a double-precision accumulator,*

$$fl(A+B) \equiv A+B+F_1 \,, \tag{2.23}$$

where

$$|F_1| \leq 2^{-t}|A+B| \,. \tag{2.24}$$

<u>Theorem 2.5</u> *If A and B are matrices conformable for matrix multiplication, then using a double-precision accumulator without floating-point accumulation of inner product,*

$$fl(AB) \equiv AB+F_2 \,, \tag{2.25}$$

where

$$(2.26) \qquad |F_2| \leq n2^{-t_1}|A|\ |B|\ ,$$

and n *is the common dimension of* A *and* B.

The proof of Theorems 2.4 and 2.5 can also be found in [17].

The following corollaries are a direct consequence of Theorem 2.4 and Theorem 2.5, respectively:

<u>Corollary 2.4.1</u>

$$(2.27) \qquad \|F_1\|_\infty \leq 2^{-t}(\|A\|_\infty+\|B\|_\infty)\ .$$

<u>Corollary 2.5.1</u>

$$(2.28) \qquad \|F_2\|_\infty \leq n2^{-t_1}\|A\|_\infty\ \|B\|_\infty\ .$$

CHAPTER III

INVERSION AND SOLUTION OF EQUATIONS BY USE OF PARTITIONED MATRICES

3.1 Schur's Identity

It is sometime advisable to use partitioned matrices when inverting fairly large sized matrices. In this section we will consider Schur's identity [9] for matrix inversion.

Let R denote a nonsingular matrix of order $2n$. We write

$$R = \left[\begin{array}{c|c} A & B \\ \hline C & D \end{array}\right], \tag{3.1}$$

where A and D are nonsingular square matrices of order r and $2n-r$, respectively. Thus B is of dimension r-by-$(2n-r)$ and C $(2n-r)$-by-r. The broken lines indicate matrix partitioning. Similarly, we partition R^{-1} in exactly the same manner as R,

$$R^{-1} = \left[\begin{array}{c|c} E & F \\ \hline G & H \end{array}\right]. \tag{3.2}$$

Since $RR^{-1} = I$, it follows that

$$(3.3)\qquad \begin{cases} AE + BG = I_{r,r} , \\ AF + BH = 0_{r,2n-r} , \\ CE + DG = 0_{2n-r,r} , \\ CF + DH = I_{2n-r,2n-r} , \end{cases}$$

where the subscripts denote the orders of the identity and zero matrices. Premultiplying the second equation of (3.3) by CA^{-1} and subtracting from the fourth, we obtain

$$(3.4)\qquad (D-CA^{-1}B)H = I_{2n-r,2n-r} ,$$

or

$$(3.5)\qquad H = \Delta^{-1} ,$$

where

$$(3.6)\qquad \Delta = (D-CA^{-1}B) .$$

From the second equation of (3.3),

$$F = -A^{-1}BH \ . \tag{3.7}$$

Similarly, the first and third equations of (3.3) give

$$G = -HCA^{-1} \ , \tag{3.8}$$

and

$$E = A^{-1} - A^{-1}BG \tag{3.9}$$

Thus we obtain

<u>Schur's Algorithm I</u> To compute R^{-1} from R in (3.1),

Step 1: Compute

(1.1) A^{-1} ,

(1.2) $A^{-1}B$,

(1.3) $C(A^{-1}B)$,

(1.4) $\Delta = D - C(A^{-1}B)$.

Step 2: Compute $H = \Delta^{-1}$.

Step 3: Compute $F = -(A^{-1}B)H$.

Step 4: Compute

(4.1) HC ,

(4.2) $G = -(HC)A^{-1}$.

Step 5: Compute

(5.1) $(A^{-1}B)G$,

(5.2) $E = A^{-1} - (A^{-1}B)G$.

In practice, all intermediate results which may be required later are saved. For example $A^{-1}B$ computed in Step 1.2 is saved to be used in Steps 1.3, 3, and 5.1.

3.2 An Error Analysis of Schur's Algorithm I

In this section, we place norm bound on the round-off error incurred in computing R^{-1} using Schur's algorithm I. Let E_i, $i=1,2,\ldots,5$ be the error at the i^{th} step and E_{ij}, the error at the Substep $i\cdot j$ of Schur's algorithm I. The subscript e and c will be used to denote exact and computed quantity, respectively.

3.2.1 A Bound for $\|E_1\|_\infty$ The computational equation for the calculation of Δ in Step 1 becomes

$$(3.10)\qquad \Delta_c \equiv [D-C\{(A_e^{-1}+E_{11})B+E_{12}\}+E_{13}]+E_{14} \ ,$$

or

$$(3.11)\qquad \Delta_c \equiv (D-CA_e^{-1}B) - CE_{11}B-CE_{12}+E_{13}+E_{14} \ .$$

But

$$\Delta_e = D - CA_e^{-1}B \ .$$

Therefore

$$(3.12) \qquad |\Delta_c - \Delta_e| = |(-CE_{11}B - CE_{12} + E_{13} + E_{14})| \ ,$$

or

$$(3.13) \qquad \| |E_1| \|_\infty = \| |CE_{11}B + CE_{12} - E_{13} - E_{14}| \|_\infty \ .$$

Using (2.5) and Property (C) in Subsection 2.1.4, it can be seen that

$$(3.14) \quad \|E_1\|_\infty \le \|C\|_\infty \|E_{11}\|_\infty \|B\|_\infty + \|C\|_\infty \|E_{12}\|_\infty + \|E_{13}\|_\infty + \|E_{14}\|_\infty \ .$$

We need satisfactory bounds for the norms of error terms on the right side of (3.14).

3.2.1.1 <u>A Bound on $\|E_{11}\|_\infty$</u> For the computed inverse A_c^{-1} of a non-singular matrix A of order n, Wilkinson [17] has shown that

$$(3.15)\qquad AA_c^{-1} - I = K ,$$

where

$$(3.16)\qquad \|K\|_\infty \leq \varepsilon g(2.005\ n^2+n^3)\|A_c^{-1}\|_\infty ,$$

with terms of order ε^2 ignored. g denotes the maximum element in magnitude at any stage in the reduction of A into upper-triangular matrix. Premultiplying (3.15) by A_e^{-1} and using (3.16), it follows that

$$(3.17)\qquad \|A_c^{-1} - A_e^{-1}\|_\infty \leq \varepsilon g(2.005\ n^2+n^3)\|A_e^{-1}\|_\infty\|A_c^{-1}\|_\infty .$$

Since the matrix A in (3.11) is of order r, we obtain from (3.17), the following norm bound on the error incurred in the calculation of inverse of A:

$$(3.18)\qquad \|E_{11}\|_\infty \leq \varepsilon g_A(2.005\ r^2+r^3)\|A_e^{-1}\|_\infty\|A_c^{-1}\|_\infty ,$$

where g_A denotes the element of maximum modulus at any stage in the reduction of A into upper-triangular matrix using Gaussian elimination.

3.2.1.2 A Bound on $\|E_{12}\|_\infty$ Let $(A_c^{-1}B)_e$ and $(A_c^{-1}B)_c$ denote the exact and computed values of $A_c^{-1}B$, respectively.

Then

$$|E_{12}| = |(A_c^{-1}B)_c - (A_c^{-1}B)_e| . \tag{3.19}$$

Using (2.28) and noting that $A_c^{-1}B$ is of dimension r-by-(2n-r), we obtain the following norm bound on the error incurred in the computation of $A_c^{-1}B$:

$$\|E_{12}\|_\infty \leq \varepsilon r \|A_c^{-1}\|_\infty \|B\|_\infty . \tag{3.20}$$

3.2.1.3 A Bound on $\|E_{13}\|_\infty$ Let $(C(A_c^{-1}B)_c)_e$ and $(C(A_c^{-1}B)_c)_c$ denote the exact and computed values of the matrix product $C(A_c^{-1}B)_c$, respectively.

Then

$$|E_{13}| = |(C(A_c^{-1}B)_c)_c - (C(A_c^{-1}B)_c)_e| . \tag{3.21}$$

Using (2.28), we derive

$$\|E_{13}\|_\infty \le \varepsilon r\|C\|_\infty\|(A_c^{-1}B)_c\|_\infty . \tag{3.22}$$

Since

$$(A_c^{-1}B)_c \equiv (A_c^{-1}B)_e+E_{12} , \tag{3.23}$$

it follows from (3.22) that

$$\|E_{13}\|_\infty \le \varepsilon r\|C\|_\infty(\|A_c^{-1}B\|_\infty+\|E_{12}\|_\infty) . \tag{3.24}$$

Thus using (3.20) in (3.24), we obtain

$$\|E_{13}\|_\infty \le \varepsilon r(1+r\varepsilon)\|C\|_\infty\|A_c^{-1}\|_\infty\|B\|_\infty . \tag{3.25}$$

3.2.1.4 <u>A Bound on $\|E_{14}\|_\infty$</u> It can be easily seen that computed value of $CA^{-1}B$ satisfies the computational equation

$$(CA^{-1}B)_c \equiv C(A_c^{-1}B+E_{12})+E_{13} . \tag{3.26}$$

Therefore, it follows from (2.27) that

$$(3.27)\qquad \|E_{14}\|_\infty \le \varepsilon\|D-\{C(A_c^{-1}B+E_{12})+E_{13}\}\|_\infty ,$$

or

$$(3.28)\quad \|E_{14}\|_\infty \le \varepsilon\|D\|_\infty+\|C\|_\infty(\|A_c^{-1}\|_\infty\|B\|_\infty+\|E_{12}\|_\infty)+\|E_{13}\|_\infty .$$

Substituting (3.20) and (3.25) in (3.28),

$$(3.29)\qquad \|E_{14}\|_\infty \le \varepsilon\{\|D\|_\infty+(1+r\varepsilon)^2\|C\|_\infty\|A_c^{-1}\|_\infty\|B\|_\infty\} .$$

Using (3.18), (3.20), (3.25) and (3.29) in (3.11), we obtain the following norm bound on E_1:

$$(3.30)\qquad \|E_1\|_\infty \le \varepsilon\|D\|_\infty+\varepsilon\{g_A(2.005\ r^2+r^3)\|A_e^{-1}\|_\infty+1+2r$$

$$+r(r+2)\varepsilon\}\|C\|_\infty\|A_c^{-1}\|_\infty\|B\|_\infty ,$$

with terms of order ε^3 ignored.

3.2.2 <u>A Bound for $\|E_2\|_\infty$</u> Let Δ_e^{-1} and Δ_c^{-1} denote the exact and computed values of $(D-CA^{-1}B)^{-1}$,

respectively. Then

$$(3.31) \qquad |E_2| = |\Delta_c^{-1} - \Delta_e^{-1}| \ .$$

Also, if $(\Delta_e+E_1)_e^{-1}$ denotes the exact inverse of computed $(D-CA^{-1}B)$, we may rewrite (3.31) as

$$(3.32) \qquad |E_2| = |\Delta_c^{-1}-(\Delta_e+E_1)_e^{-1}+(\Delta_e+E_1)_e^{-1}-\Delta_e^{-1}| \ .$$

It immediately follows from (3.32) that

$$(3.33) \qquad \|E_2\|_\infty \le \|\Delta_c^{-1}-(\Delta_e+E_1)_e^{-1}\|_\infty+\|(\Delta_e+E_1)_e^{-1}-\Delta_e^{-1}\|_\infty \ .$$

Using (3.17) in (3.34),

$$(3.34) \qquad \|E_2\|_\infty \le \varepsilon\ g_\Delta\{2.005(2n-r)^2+(2n-r)^3\}\|(\Delta_e+E_1)^{-1}\|_\infty\|\Delta_c^{-1}\|_\infty$$

$$+\|(\Delta_e+E_1)_e^{-1}-\Delta_e^{-1}\|_\infty \ ,$$

where g_Δ denotes the element of maximum modulus in the reduction of Δ into upper-triangular matrix using Gaussian elimination. The following theorem is a result of Property (F) in Subsection 2.1.4 and (3.34):

<u>Theorem 3.1</u> *Let* E_2 *denote the matrix of round-off errors in the calculation of* H *using Schur's algorithm I (see Section 3.1), then if* $\|\Delta_e^{-1}\|_\infty \|E_1\|_\infty < 1$,

$$(3.35) \qquad \|E_2\|_\infty \le \left[\varepsilon g_\Delta \{2.005(2n-r)^2 + (2n-r)^3\} \|\Delta_c^{-1}\|_\infty + \|\Delta_e^{-1}\|_\infty \|E_1\|_\infty \right] \frac{\|\Delta_e^{-1}\|_\infty}{1-\|\Delta_e^{-1}\|_\infty \|E_1\|_\infty},$$

where $\|E_1\|_\infty$ *is defined by (3.30).*

3.2.3 <u>A Bound for $\|E_3\|_\infty$</u> The computational equation for the calculation of F becomes

$$(3.36) \qquad F_c \equiv -[\{(A_e^{-1}+E_{11})B+E_{12}\}(H_e+E_2)+E_{31}],$$

where E_{31} denotes the error in matrix multiplication $(A_c^{-1}B)_c H_c$. Since $\Delta_c^{-1} \equiv H_e+E_2$, we obtain from (3.36),

$$(3.37) \qquad |E_3| = |A_e^{-1}BE_2+(E_{11}B+E_{12})\Delta_c^{-1}+E_{31}|,$$

or

$$(3.38) \qquad \|E_3\|_\infty \le \|A_e^{-1}\|_\infty \|B\|_\infty \|E_2\|_\infty + (\|E_{11}\|_\infty \|B\|_\infty + \|E_{12}\|_\infty)\|\Delta_c^{-1}\|_\infty + \|E_{31}\|_\infty .$$

3.2.3.1 <u>A Bound on $\|E_{31}\|_\infty$</u> Let $((A_c^{-1}B)_c\Delta_c^{-1})_e$ and $((A_c^{-1}B)_c\Delta_c^{-1})_c$ denote the exact and computed values of $(A_c^{-1}B)_c\Delta_c^{-1}$, respectively. Then

$$(3.39)\qquad |E_{31}| = |((A_c^{-1}B)_c\Delta_c^{-1})_c - ((A_c^{-1}B)_c\Delta_c^{-1})_e| .$$

Using (2.28),

$$(3.40)\qquad \|E_{31}\|_\infty \le \varepsilon(2n-r)(\|A_c^{-1}\|_\infty\|B\|_\infty+\|E_{12}\|_\infty)\|\Delta_c^{-1}\|_\infty .$$

Substituting (3.20) in (3.40),

$$(3.41)\qquad \|E_{31}\|_\infty \le \varepsilon(2n-r)(1+r\varepsilon)\|A_c^{-1}\|_\infty\|B\|_\infty\|\Delta_c^{-1}\|_\infty .$$

Using (3.18), (3.20) and (3.41) in (3.38),

$$(3.42)\qquad \|E_3\|_\infty \le \|A_e^{-1}\|_\infty\|B\|_\infty\|E_2\|_\infty+\varepsilon\{g_A(2.005\ r^2+r^3)\|A_e^{-1}\|_\infty$$

$$+2n+r(2n-r)\varepsilon\}\|A_c^{-1}\|_\infty\|B\|_\infty\|\Delta_c^{-1}\|_\infty .$$

Hence, we state

<u>Theorem 3.2</u> *Let* E_3 *denote the matrix of round-off errors in the calculation of* F *using Schur's Algorithm I (see Section 3.1), then*

$$(3.43) \quad \|E_3\|_\infty \le \|A_e^{-1}\|_\infty \|B\|_\infty \|E_2\|_\infty + \varepsilon\{g_A(2.005\ r^2+r^3)\|A_e^{-1}\|_\infty$$

$$+2n+r(2n-r)\varepsilon\}\|A_c^{-1}\|_\infty \|B\|_\infty \|\Delta_c^{-1}\|_\infty ,$$

where $\|E_2\|_\infty$ *is defined by (3.35).*

3.2.4 <u>A Bound for $\|E_4\|_\infty$</u> The computational equation for the calculation of G is

$$(3.44) \qquad G_c \equiv -[\{(H_e+E_2)C+E_{41}\}(A_e^{-1}+E_{11})+E_{42}]$$

Since $H_e \equiv \Delta_e^{-1}$ and $A_c^{-1} \equiv A_e^{-1}+E_{11}$, therefore

$$(3.45) \qquad |E_4| = |\Delta_e^{-1}CE_{11}+(E_2C+E_{41})A_c^{-1}+E_{42}| ,$$

or

$$(3.46) \quad \|E_4\|_\infty \le \|\Delta_e^{-1}\|_\infty \|C\|_\infty \|E_{11}\|_\infty + (\|E_2\|_\infty \|C\|_\infty + \|E_{41}\|_\infty)\|A_c^{-1}\|_\infty + \|E_{42}\|_\infty .$$

3.2.4.1 Bounds on $\|E_{41}\|_\infty$ and $\|E_{42}\|_\infty$ A norm bound on the error incurred in the matrix multiplication H_cC can be derived using (2.28). Thus

$$\|E_{41}\|_\infty \leq \varepsilon(2n-r)\|\Delta_c^{-1}\|_\infty\|C\|_\infty . \tag{3.47}$$

Similarly, a norm bound on the error incurred in the matrix multiplication $(H_cC)A_c^{-1}$ follows from Subsection 3.2.1.3. Thus

$$\|E_{42}\|_\infty \leq \varepsilon r(\|\Delta_c^{-1}\|_\infty\|C\|_\infty+\|E_{41}\|_\infty)\|A_c^{-1}\|_\infty . \tag{3.48}$$

Using (3.47),

$$\|E_{42}\|_\infty \leq \varepsilon r\{1+(2n-r)\varepsilon\}\|\Delta_c^{-1}\|_\infty\|C\|_\infty\|A_c^{-1}\|_\infty . \tag{3.49}$$

The following theorem is a result of (3.18), (3.46), (3.47) and (3.49):

Theorem 3.3 *Let E_4 denote the matrix of round-off errors in the calculation of G using Schur's Algorithm I (see Section 3.1), then*

$$(3.50)\quad \|E_4\|_\infty \leq [\|E_2\|_\infty + \varepsilon\{g_A(2.005\ r^2+r^3)\|A_e^{-1}\|_\infty\|\Delta_e^{-1}\|_\infty\}$$

$$+\ \varepsilon\{2n+r(2n-r)\varepsilon\}\|\Delta_c^{-1}\|_\infty]\|A_c^{-1}\|_\infty\|C\|_\infty\ ,$$

where $\|E_2\|_\infty$ *is given by (3.35).*

3.2.5 <u>A Bound on $\|E_5\|_\infty$</u> The computational equation for the calculation of E is

$$(3.51)\quad E_c \equiv [A_e^{-1}+E_{11} - \{(A_e^{-1}+E_{11})B+E_{12}\}(G_e+E_4)+E_{51}+E_{52}\ ,$$

or

$$(3.52)\quad |E_5| = |E_{11}-A_e^{-1}BE_4+(E_{11}B+E_{12})G_c+E_{51}+E_{52}|\ ,$$

or

$$(3.53)\quad \|E_5\|_\infty \leq \|E_{11}\|_\infty+\|A_e^{-1}\|_\infty\|B\|_\infty\|E_4\|_\infty+(\|E_{11}\|_\infty\|B\|_\infty+\|E_{12}\|_\infty)\|G_c\|_\infty$$

$$+\ \|E_{51}\|_\infty+\|E_{52}\|_\infty\ .$$

We now proceed to place norm bounds on the matrices E_{51} and E_{52}.

3.2.5.1 <u>Bounds on $\|E_{51}\|_\infty$ and $\|E_{52}\|_\infty$</u> Norm bounds for the matrices of round-off errors E_{51} and E_{52} incurred in the calculation of $(A^{-1}B)G$ and $A^{-1} - (A^{-1}B)G$, respectively

can be obtained by using an analysis analogous to that in Subsection 3.2.1. We derive

$$(3.54)\qquad \|E_{51}\|_\infty \leq \varepsilon(2n-r)(1+r\varepsilon)\|A_c^{-1}\|_\infty\|B\|_\infty\|G_c\|_\infty \,,$$

and

$$(3.55)\qquad \|E_{52}\|_\infty \leq \varepsilon[1+(1+r\varepsilon)\{1+(2n-r)\varepsilon\}\|B\|_\infty\|G_c\|_\infty]\|A_c^{-1}\|_\infty \,.$$

The following theorem is a result of (3.18), (3.20), (3.53), (3.54) and (3.55):

<u>Theorem 3.4</u> *Let E_5 denote the matrix of round-off errors in the calculation of E using Schur's algorithm I (see Section 3.1), then if terms of order ε^3 are neglected,*

$$(3.56)\qquad \|E_5\|_\infty \leq \|A_e^{-1}\|_\infty\|B\|_\infty\|E_4\|_\infty+\varepsilon\{1+g_A(2.005\ r^2+r^3)\|A_e^{-1}\|_\infty$$

$$\times\ \|A_c^{-1}\|_\infty+\varepsilon\{g_A(2.005\ r^2+r^3)\|A_e^{-1}\|_\infty+1+2n(1+\varepsilon)$$

$$+\ r(2n-r)\varepsilon\}\|A_c^{-1}\|_\infty\|B\|_\infty\|G_c\|_\infty \,,$$

where $\|E_4\|_\infty$ is defined by (3.50).

3.3 A Bound for the Computed Inverse of R

Let R_e^{-1} and R_c^{-1} denote the exact and computed inverse of R, respectively. Then the computational equation for the calculation of R^{-1} becomes

$$(3.57)\qquad R_c^{-1} \equiv \left[\begin{array}{c|c} E_e+E_5 & F_e+E_3 \\ \hline G_e+E_4 & H_e+E_2 \end{array}\right],$$

where E_2, E_3, E_4 and E_5 are given by (3.35), (3.43) (3.50) and (3.56), respectively.

Since
$$R_e^{-1} = \left[\begin{array}{c|c} E_e & F_e \\ \hline G_e & H_e \end{array}\right],$$

it follows from (3.57) that

$$(3.58)\qquad |R_c^{-1} - R_e^{-1}| = \left[\begin{array}{c|c} |E_5| & |E_3| \\ \hline |E_4| & |E_2| \end{array}\right].$$

Hence,

$$(3.59)\qquad \|R_c^{-1} - R_e^{-1}\|_\infty \le \|E_2\|_\infty+\|E_3\|_\infty+\|E_4\|_\infty+\|E_5\|_\infty ,$$

where $\|E_2\|_\infty$, $\|E_3\|_\infty$, $\|E_4\|_\infty$ and $\|E_5\|_\infty$ are bounded by (3.35), (3.43), (3.50) and (3.56), respectively.

3.4 A Smaller Bound for $\|R_c^{-1} - R_e^{-1}\|_\infty$

A smaller bound on $\|R_c^{-1} - R_e^{-1}\|_\infty$ can be placed by using the definition of ∞-norm. Since the maximum row-sum of error matrix on the right side of (3.58) can occur either in matrix $[E_5 \mid E_3]$ or $[E_4 \mid E_2]$, it immediately follows that at least one of the following is true:

$$(3.60) \qquad \|R_c^{-1} - R_e^{-1}\|_\infty \le \begin{cases} \|E_2\|_\infty + \|E_4\|_\infty \,, \\ \\ \|E_3\|_\infty + \|E_5\|_\infty \,. \end{cases}$$

This bound is at least as small as that given by (3.59).

3.5 Solution of System of Equations

Let

$$(3.61) \qquad Rx = b \,,$$

be a system of linear algebraic equations of order $2n$ with nonsingular coefficient matrix R. Then, we can represent (3.61) in the form

$$(3.62) \qquad \left[\begin{array}{c|c} A & B \\ \hline C & D \end{array}\right] \begin{bmatrix} x_1 \\ \hline x_2 \end{bmatrix} = \begin{bmatrix} b_1 \\ \hline b_2 \end{bmatrix} \,,$$

where A and D are nonsingular square matrices of order r and $2n-r$, respectively. The matrix B is of dimension r-by-$(2n-r)$ and C of dimension $(2n-r)$-by-r. Thus the vectors x_1 and b_1 are of order r and vectors x_2 and b_2 of order $2n-r$, each. The broken lines indicate matrix partitioning. Equating terms on both sides of (3.62),

$$\left\{\begin{aligned} Ax_1 + Bx_2 &= b_1 , \\ Cx_1 + Dx_2 &= b_2 . \end{aligned}\right. \tag{3.63}$$

Eliminating x_1 from (3.63),

$$(D-CA^{-1}B)x_2 = b_2-CA^{-1}b_1 . \tag{3.64}$$

Thus, if the matrix $(D-CA^{-1}B)$ is nonsingular, the system (3.64) can be solved for x_2 using Gaussian elimination. The solution vector x_2 can not, however, be substituted in the second equation to determine x_1, since the matrix C is rectangular for all $r \neq n$. This suggests that, in such case, we resort to a method of evolving a system of equations similar to (3.64). Thus eliminating x_2 from (3.63),

$$(A-BD^{-1}C)x_1 = b_1-BD^{-1}b_2 \quad . \tag{3.65}$$

Hence, we obtain

<u>Schur's Algorithm II</u> To compute $x=(x_1,x_2)$ from (3.63) when $r \neq n$

Step 1: Compute

(1.1) A^{-1} ,

(1.2) CA^{-1} ,

(1.3) $(CA^{-1})B$.

Step 2: Compute

(2.1) $(CA^{-1})b_1$,

(2.2) $b_2-(CA^{-1})b_1$.

Step 3: Compute x_2 from (3.64).

Step 4: Compute

(4.1) D^{-1} ,

(4.2) BD^{-1} ,

(4.3) $(BD^{-1})C$.

(4.4) $A - (BD^{-1})C$.

Step 5: Compute

(5.1) $(BD^{-1})b_2$,

(5.2) $b_1-(BD^{-1})b_2$.

Step 6: Compute x_1 from (3.65).

When the matrices A, B, C and D are square, the last three steps in the above algorithm are replaced by

Step 4: Compute

(4.1) Dx_2 ,

(4.2) b_2-Dx_2 .

Step 5: Compute x_1 from $Cx_1 = b_2-Dx_2$.

3.6 An Error Analysis of Schur's Algorithm II with $r \neq n$

We now proceed to place norm bounds on the round-off error incurred in computing the solution of the system of equations (3.61) using Schur's Algorithm II. We will use the same notations as in Section 3.2.

3.6.1 Bounds for $\|E_1\|_\infty$ and $\|E_2\|_\infty$

A norm bound for the error in computing $D-CA^{-1}B$ has already been obtained in Subsection 3.2.1. Although the algorithm for calculating $D-CA^{-1}B$ in Section 3.1 is slightly different from that in Section 3.6, it can be easily shown that round-off error bounds are exactly the same. Therefore,

$$\|E_1\|_\infty \le \varepsilon\|D\|_\infty+\varepsilon\{g_A(2.005\ r^2+r^3)\|A_e^{-1}\|_\infty+1+2r + r(r+2)\varepsilon\}\|C\|_\infty\|A_c^{-1}\|_\infty\|B\|_\infty\ , \tag{3.66}$$

with terms of order ε^3 ignored.

Similarly, a norm bound for E_2 can be derived from (3.66) by replacing D by b_2, and B by b_1,

$$(3.67) \qquad \|E_2\|_\infty \leq \varepsilon\|b_2\|_\infty+\varepsilon\{g_A(2.005\ r^2+r^3)\|A_e^{-1}\|_\infty+1+2r + r(r+2)\varepsilon\}\|C\|_\infty\|A_c^{-1}\|_\infty\|b_1\|_\infty .$$

3.6.2 <u>A Bound for $\|E_3\|_\infty$</u> Wilkinson [17] has shown that the computed solution of $Ax=b$, is an exact solution of the perturbed system $(A+K)x=b$,

where

$$(3.68) \qquad \|K\|_\infty \leq \varepsilon g(2.005\ n^2+n^3) .$$

g denotes the element of maximum modulus at all stages in the reduction of A using Gaussian elimination. If $x_e^{(2)}$ and $x_c^{(2)}$ denote the exact and computed solution of (3.64), it follows that $x_c^{(2)}$ is an exact solution of the perturbed system

$$(3.69) \qquad (\Delta_c+K_1)x_c^{(2)} = b_e^{(2)}+E_2 ,$$

where $b_e^{(2)}$ denotes the exact value of $b_2 - CA^{-1}b_1$, and

(3.70) $$\|K_1\|_\infty \leq \varepsilon g_\Delta \{2.005(2n-r)^2 + (2n-r)^3\} .$$

g_Δ denotes the element of maximum magnitude at all stages in the reduction of Δ into upper-triangular matrix using Gaussian elimination. Since $\Delta_c = \Delta_e + E_1$ and $\Delta_e x_e^{(2)} = b_e^{(2)}$, we obtain from (3.69),

(3.71) $$|x_c^{(2)} - x_e^{(2)}| = |\Delta_e^{-1} E_2 - \Delta_e^{-1}(E_1 + K_1) x_c^{(2)}| ,$$

or

(3.72) $$\|E_3\|_\infty \leq \|E_2\|_\infty \|\Delta_e^{-1}\|_\infty + (\|E_1\|_\infty + \|K_1\|_\infty)\|\Delta_e^{-1}\|_\infty \|x_c^{(2)}\|_\infty ,$$

since $$|E_3| = |x_c^{(2)} - x_e^{(2)}| .$$

3.6.3 <u>Bounds on $\|E_4\|_\infty$, $\|E_5\|_\infty$ and $\|E_6\|_\infty$</u> A reference to equations (3.66), (3.67), (3.70) and (3.72) shows that norm bounds for E_4, E_5 and E_6 can be obtained merely by replacing D by A, C by B, A by D, B by C, Δ_e^{-1} by δ_e^{-1}, b_2 by b_1, b_1 by b_2, $(2n-r)$ by r, r by $(2n-r)$, $x_c^{(2)}$ by $x_c^{(1)}$, E_1 by E_4, E_2 by E_5, E_3 by E_6 and K_1 by K_2 .

Therefore,

$$(3.73)\qquad \|E_6\|_\infty \leq \|E_5\|_\infty \|\delta_e^{-1}\|_\infty + (\|E_4\|_\infty + \|K_2\|_\infty)\|\delta_e^{-1}\|_\infty \|x_c^{(1)}\|_\infty ,$$

where

$$(3.74)\qquad \|E_4\|_\infty \leq \varepsilon\|A\|_\infty + \varepsilon[g_D\{2.005(2n-r)^2+(2n-r)^3\}\|D_e^{-1}\|_\infty$$

$$+1+2(2n-r)+(2n-r)(2n-r+2)\varepsilon]\|C\|_\infty \|D_c^{-1}\|_\infty \|B\|_\infty ,$$

$$(3.75)\qquad \|E_5\|_\infty \leq \varepsilon\|b_1\|_\infty + \varepsilon[g_D\{2.005(2n-r)^2+(2n-r)^3\}\|D_e^{-1}\|_\infty$$

$$+1+2(2n-r)+(2n-r)(2n-r+2)\varepsilon]\|B\|_\infty \|D_c^{-1}\|_\infty \|b_2\|_\infty ,$$

and

$$(3.76)\qquad \|K_2\|_\infty \leq \varepsilon\ g_\delta(2.005\ r^2+r^3) .$$

δ_e and $x_c^{(1)}$ denote the exact value of $A-BD^{-1}C$ and the computed solution of (3.65), respectively.

The following theorem is a result of (3.72) and (3.73):

Theorem 3.5 *Let x_e and x_c denote the exact and computed solution of (3.61) using Schur's Algorithm II (see Section 3.5), then at least one of the following is true:*

$$(3.77)\qquad \|x_c-x_e\|_\infty \leq \begin{cases} \|E_2\|_\infty\|\Delta_e^{-1}\|_\infty+(\|E_1\|_\infty+\|K_1\|_\infty)\|\Delta_e^{-1}\|_\infty\|x_c^{(2)}\|_\infty\ , \\ \|E_5\|_\infty\|\delta_e^{-1}\|_\infty+(\|E_4\|_\infty+\|K_2\|_\infty)\|\delta_e^{-1}\|_\infty\|x_c^{(1)}\|_\infty\ , \end{cases}$$

where norm bounds on matrices E_1, E_2, E_4, E_5, K_1 and K_2 are given by (3.66), (3.67), (3.74), (3.75), (3.70) and (3.76), respectively.

3.7 An Error Analysis of Schur's Algorithm II with r=n

In the following we propose to place norm bounds on the matrices of round-off errors E_4 and E_5 incurred in computing the last two steps of Schur's Algorithm II with r=n (see Section 3.5). The error analysis of the first three steps of this algorithm has already been given in Section 3.6.

3.7.1 A Bound on $\|E_4\|_\infty$

The computational equation for the calculation of b_2-Dx_2 is

$$(3.78)\qquad b_2-Dx_2 \equiv \{b_2-Dx_c^{(2)}+E_{41}\}+E_{42}\ ,$$

or

$$(3.79)\qquad b_2 - Dx_2 \equiv \{b_2 - D(x_e^{(2)} + x_c^{(2)} - x_e^{(2)}) + E_{41}\} + E_{42} \ ,$$

from which it follows that

$$(3.80)\qquad \|E_4\|_\infty \leq \|D\|_\infty \|x_c^{(2)} - x_e^{(2)}\|_\infty + \|E_{41}\|_\infty + \|E_{42}\|_\infty \ .$$

We now need satisfactory norm bounds for the matrices E_{41} and E_{42}. A bound for $\|x_c^{(2)} - x_e^{(2)}\|_\infty$ is given by (3.72), since $E_3 = x_c^{(2)} - \dot{x}_e^{(2)}$.

3.7.1.1 Bounds for $\|E_{41}\|_\infty$ and $\|E_{42}\|_\infty$

Using (2.27) and (2.28),

$$(3.81)\qquad \|E_{41}\|_\infty \leq \varepsilon\ n\|D\|_\infty \|x_c^{(2)}\|_\infty \ ,$$

and

$$(3.82)\qquad \|E_{42}\|_\infty \leq \varepsilon\|b_2 - Dx_c^{(2)} + E_{41}\|_\infty \ .$$

Using (3.81) in (3.82),

$$(3.83)\qquad \|E_{42}\|_\infty \leq \varepsilon\{\|b_2\|_\infty + (1+n\varepsilon)\|D\|_\infty \|x_c^{(2)}\|_\infty\} \ .$$

Substituting (3.81) and (3.83) into (3.80), we obtain the following norm bound for the matrix of round-off errors incurred in computing $b_2 - Dx_2$:

$$(3.84) \quad \|E_4\|_\infty \leq \varepsilon[\|b_2\|_\infty + (n+1+n\varepsilon)\|D\|_\infty \|x_c^{(2)}\|_\infty] + \|D\|_\infty \|x_c^{(2)} - x_e^{(2)}\|_\infty .$$

3.7.2 <u>A Bound on $\|E_5\|_\infty$</u> Using a method analogous to that in Subsection 3.6.2, we derive that

$$(3.85) \qquad \|E_5\|_\infty \leq (\|K_2\|_\infty \|x_c^{(1)}\|_\infty + \|E_4\|_\infty)\|C^{-1}\|_\infty ,$$

where

$$(3.86) \qquad \|K_2\|_\infty \leq \varepsilon\, g_C(2.005\, n^2 + n^3) .$$

A bound for $\|E_4\|_\infty$ is given by (3.84).

The following theorem is a result of (3.72) and (3.85):

<u>Theorem 3.6</u> *Let x_e and x_c denote the exact and computed solution of (3.61), then if the matrices A, B, C, D are square, at least one of the following is*

true:

$$(3.87) \quad \|x_c - x_e\|_\infty \le \begin{cases} \|E_2\|_\infty \|\Delta_e^{-1}\|_\infty + (\|E_1\|_\infty + \|K_1\|_\infty)\|\Delta_e^{-1}\|_\infty \|x_c^{(2)}\|_\infty , \\ (\|E_4\|_\infty + \|K_2\|_\infty \|x_c^{(1)}\|_\infty)\|C^{-1}\|_\infty , \end{cases}$$

where norm bounds on E_1, E_2, E_4, K_1 *and* K_2 *are given by (3.66), (3.67), (3.74), (3.70) and (3.76), respectively, with* r *replaced by* n.

3.8 Comparison of Norm Bounds

A reasonable theoretical comparison of the norm bounds on the round-off errors incurred, in computing the inverse of a matrix and solution of a system of linear algebraic equations, using Gaussian elimination and Schur's Algorithms (see Section 3.1 and Section 3.5), can not be done due to the nature of terms involved in the latter. However, a comparison of the norm bounds when $r=n$, can be made.

Without loss of generality, we can assume that elements of the matrix R have been scaled such that

$$(3.88) \qquad |r_{ij}| \le 1 .$$

Therefore,

(3.89) $$\|A\|_\infty, \|B\|_\infty, \|C\|_\infty \text{ and } \|D\|_\infty \leq n .$$

From the definition of ∞-norm of a matrix and (3.57) it follows that

(3.90) $$\|\Delta_c^{-1}\|_\infty \leq \|R_c^{-1}\|_\infty ,$$

(3.91) $$\|\Delta_e^{-1}\|_\infty \leq \|R_e^{-1}\|_\infty ,$$

(3.92) $$\|G_c\|_\infty \leq \|R_c^{-1}\|_\infty ,$$

(3.93) $$\|E_c\|_\infty \leq \|R_c^{-1}\|_\infty ,$$

and

(3.94) $$\|E_e\|_\infty \leq \|R_e^{-1}\|_\infty .$$

Also, since $E = A^{-1}-(A^{-1}B)G$,

(3.95) $$\|A_e^{-1}\|_\infty \leq \|R_e^{-1}\|_\infty ,$$

and

(3.96) $$\|A_c^{-1}\|_\infty \leq \|R_c^{-1}\|_\infty .$$

Using inequalities (3.89) to (3.96) in (3.30), (3.35), (3.42), (3.50), (3.56) and (3.60), it follows that the dominant term in the expression for round-off error incurred in computing the inverse using Schur's algorithm I (see Section 3.1) is $O(n^{\alpha})$ $\alpha \geq 5$, whereas the corresponding error term using Gaussian elimination is $O(n^3)$. The latter follows from (3.17) by replacing A by R and n by 2n. Similarly, it can be shown that the dominant term in the expression for round-off error incurred in computing the solution of a system of linear equations using Schur's algorithm II (see Section 3.5) and Gaussian elimination is $O(n^{\alpha})$, $\alpha \geq 5$, and $O(n^3)$, respectively. However, the actual errors depend on the nature of the elements of the matrices considered.

CHAPTER IV

BLOCK-SYMMETRIC MATRICES

4.1 Inversion by Use of Schur's Identity

Let R be a nonsingular block-symmetric matrix of order $2n$. We write

$$R = \left[\begin{array}{c|c} A & B \\ \hline B & A \end{array}\right], \tag{4.1}$$

where A and B are square matrices of order n. Similarly we partition R^{-1} in exactly the same manner as R,

$$R^{-1} = \left[\begin{array}{c|c} E & F \\ \hline F & E \end{array}\right], \tag{4.2}$$

It can be easily shown by equating the terms on both sides of $RR^{-1} = I$, that

$$E = (A-BA^{-1}B)^{-1}, \tag{4.3}$$

$$F = (B-AB^{-1}A)^{-1}, \tag{4.4}$$

provided the matrices A, B, $(A-BA^{-1}B)$ and $(B-AB^{-1}A)$ are nonsingular. Thus we obtain

Schur's Algorithm III To compute R^{-1} from R in (4.1),

Step 1: Compute

(1.1) A^{-1} ,

(1.2) $A^{-1}B$,

(1.3) $B(A^{-1}B)$,

(1.4) $\Delta = A-B(A^{-1}B)$.

Step 2: Compute $E = \Delta^{-1}$.

Step 3: Compute

(3.1) B^{-1} ,

(3.2) $B^{-1}A$,

(3.3) $A(B^{-1}A)$,

(3.4) $\delta = B-A(B^{-1}A)$.

Step 4: Compute $F = \delta^{-1}$.

4.2 An Error Analysis of Schur's Algorithm III

In this section we place norm bounds on the round-off errors incurred at each step in the computation of R^{-1} using Schur's Algorithm III. We will use the same notations as in Section 3.2.

4.2.1 Bounds on $\|E_1\|_\infty$, $\|E_2\|_\infty$, $\|E_3\|_\infty$ and $\|E_4\|_\infty$

The error analysis of the inversion of a block-symmetric matrix using Schur's Algorithm III is similar to that

developed in Section 3.2. Bounds for $\|E_1\|_\infty$ and $\|E_2\|_\infty$ follow from (3.30) and (3.35), respectively, by replacing r by n, C by B and D by A. We have therefore

$$(4.5) \quad \|E_1\|_\infty \leq \varepsilon\|A\|_\infty+\varepsilon\{g_A(2.005\ n^2+n^3)\|A_e^{-1}\|_\infty+1+2n + n(n+2)\varepsilon\}\|B\|_\infty\|A_c^{-1}\|_\infty\|B\|_\infty ,$$

with terms of order ε^3 ignored, and

$$(4.6) \quad \|E_2\|_\infty \leq \frac{\{\varepsilon g_\Delta(2.005\ n^2+n^3)\|\Delta_c^{-1}\|_\infty+\|\Delta_e^{-1}\|_\infty\|E_1\|_\infty\}\|\Delta_e^{-1}\|_\infty}{1-\|\Delta_e^{-1}\|_\infty\|E_1\|_\infty} .$$

Bounds for $\|E_3\|_\infty$ and $\|E_4\|_\infty$ follow similarly from (3.30) and (3.35), respectively, by replacing r by n, A by B, B by A, C by A, D by B, Δ by δ and E_1 by E_3, hence,

$$(4.7) \quad \|E_3\|_\infty \leq \varepsilon\|B\|_\infty+\varepsilon\{g_B(2.005\ n^2+n^3)\|B_e^{-1}\|_\infty+1+2n+n(n+2)\varepsilon\} \times \|A\|_\infty\|B_c^{-1}\|_\infty\|A\|_\infty ,$$

with terms of order ε^3 ignored, and

$$(4.8) \qquad \|E_4\| \leq \frac{\{\varepsilon g_\delta(2.005\ n^2+n^3)\|\delta_c^{-1}\|_\infty+\|\delta_e^{-1}\|_\infty\|E_3\|_\infty\}\|\delta_e^{-1}\|_\infty}{1-\|\delta_e^{-1}\|_\infty\|E_3\|_\infty}.$$

The following theorem is a result of (4.1), (4.2), (4.6) and (4.8):

<u>Theorem 4.1</u> *Let* R_e^{-1} *and* R_c^{-1} *denote the exact and computed inverse of a nonsingular block-symmetric matrix* $R = \left[\begin{array}{c|c} A & B \\ \hline B & A \end{array}\right]$ *of order* $2n$, *where* A *and* B *are square matrices of order* n. *If the matrices* A, B, $A-BA^{-1}B$ *and* $B-AB^{-1}A$ *are nonsingular, then the norm bound for the round-off errors incurred in computing* R^{-1} *using Schur's Algorithm III (see Section 4.1) is given by*

$$(4.9) \qquad \|R_c^{-1}-R_e^{-1}\|_\infty \leq \|E_2\|_\infty+\|E_4\|_\infty ,$$

where $\|E_2\|_\infty$ *and* $\|E_4\|_\infty$ *are given by (4.6) and (4.8), respectively.*

4.3 <u>Inversion by Use of Charmonman's Algorithm</u>

A method for computing the inverse of a block-symmetric matrix has been proposed by Charmonman [1]. This method is more efficient in the sense that it results in saving

of computer storage and the number of arithmetic operations. The algorithm given by Charmonman [1] can be devised by use of the following theorem:

Theorem 4.2 *The inverse of* $R = \left[\begin{array}{c|c} A & B \\ \hline B & A \end{array}\right]$ *is* $\left[\begin{array}{c|c} E & F \\ \hline F & E \end{array}\right]$,

where

$$(4.10)\qquad \begin{cases} E = 0.5(P^{-1}+Q^{-1}) \ , \\ F = 0.5(P^{-1}-Q^{-1}) \ , \\ P = A+B \ , \\ Q = A-B \ . \end{cases}$$

Proof: The proof of the first part of the theorem follows immediately from Schur's Algorithm I (see Section 3.1) by replacing C by B and D by A. The second part of the theorem can be proved as follows: We may write $R^{-1}R \equiv I$ as

$$(4.11)\qquad \left[\begin{array}{c|c} A & B \\ \hline B & A \end{array}\right] \left[\begin{array}{c|c} E & F \\ \hline F & E \end{array}\right] \equiv I_{2n,2n} \ ,$$

where $I_{2n,2n}$ is an identity matrix of order $2n$. Equating terms on both sides of (4.11), we obtain

$$AE + BF = I_{n,n} , \tag{4.12}$$

and

$$BE + AF = O_{n,n} . \tag{4.13}$$

Adding (4.12) and (4.13), and premultiplying by $(A+B)^{-1}$ on both sides, we derive

$$E+F = (A+B)^{-1} . \tag{4.14}$$

Similarly, subtracting (4.13) from (4.12), and premultiplying by $(A-B)^{-1}$ on both sides, we obtain

$$E-F = (A-B)^{-1} . \tag{4.15}$$

The result in (4.10) follows by solving (4.14) and (4.15) for E and F.

Q.E.D.

Thus we obtain

<u>Charmonman's Algorithm I</u> To compute R^{-1} from R in (4.1),

Step 1: Compute $P = A+B$.

Step 2: Compute $Q = P-2B$.

Step 3: Compute P^{-1} .

Step 4: Compute Q^{-1} .

Step 5: Compute $E = 0.5(P^{-1}+Q^{-1})$.

Step 6: Compute $F = E-Q^{-1}$.

Multiplication by 2 and 0.5 are done by shifting without any round-off error.

4.4 An Error Analysis of Charmonman's Algorithm I

We now proceed to place upper norm bound on the round-off error incurred in computing R^{-1} using the above algorithm. We shall again use the same notations as in Section 3.2.

4.4.1 Bounds on $\|E_1\|_\infty$ and $\|E_2\|_\infty$

Let P_e' and P_c' denote the exact and computed value of P, then

$$|E_1| = |P_c'-P_e'| . \tag{4.16}$$

It follows from (2.27) that

$$\|E_1\|_\infty \le \varepsilon(\|A\|_\infty+\|B\|_\infty) . \tag{4.17}$$

Similarly, if Q_e' and Q_c' denote the exact and computed values of Q,

$$|E_2| = |Q_c'-Q_e'| . \tag{4.18}$$

From the computational equation for the calculation of Q,

$$Q_c' \equiv P_e'+E_1-2B+E_{21} , \tag{4.19}$$

and (4.18), we have

$$\|E_2\|_\infty = \|E_1+E_{21}\|_\infty , \tag{4.20}$$

or

$$\|E_2\|_\infty \le \|E_1\|_\infty+\|E_{21}\|_\infty . \tag{4.21}$$

4.4.1.1 <u>A Bound on $\|E_{21}\|_\infty$</u> Since E_{21} is the matrix of round-off errors incurred due to the addition of computed P and $-2B$, it follows from (2.27) that $\|E_{21}\|_\infty \leq \varepsilon\|P_e'+E_1-2B\|_\infty$. Using (4.17) ,

$$\|E_{21}\|_\infty \leq \varepsilon\{\|A\|_\infty+3\|B\|_\infty+\varepsilon(\|A\|_\infty+\|B\|_\infty)\} . \tag{4.22}$$

Substituting (4.17) and (4.22) into (4.21) give

$$\|E_2\|_\infty \leq 2\varepsilon(\|A\|_\infty+2\|B\|_\infty) , \tag{4.23}$$

with terms of order ε^2 ignored.

4.4.2 <u>Bounds on $\|E_3\|_\infty$ and $\|E_4\|_\infty$</u> A reference to the error analysis in Subsection 3.2.2 shows that bounds for $\|E_3\|_\infty$ and $\|E_4\|_\infty$ can be obtained from (3.35) by replacing r by n, Δ by P and Q, respectively. Thus

$$\|E_3\|_\infty \leq \frac{\varepsilon\{g_p(2.005\ n^2+n^3)\|P_c^{-1}\|_\infty+\|P_e^{-1}\|_\infty\|E_1\|_\infty\}\|P_e^{-1}\|_\infty}{1-\|P_e^{-1}\|_\infty\|E_1\|_\infty} , \tag{4.24}$$

and

$$(4.25) \qquad \|E_4\|_\infty \le \frac{\varepsilon\{g_Q(2.005\ n^2+n^3)\|Q_c^{-1}\|_\infty+\|Q_e^{-1}\|_\infty\|E_2\|_\infty\}\|Q_e^{-1}\|_\infty}{1-\|Q_e^{-1}\|_\infty\|E_2\|_\infty},$$

provided

$$(4.26) \qquad \|P_e^{-1}\|_\infty\|E_1\|_\infty < 1 ,$$

and

$$(4.27) \qquad \|Q_e^{-1}\|_\infty\|E_2\|_\infty < 1 .$$

Norm bounds on E_1 and E_2 are given by (4.17) and (4.23), respectively.

4.4.3 A Bound on $\|E_5\|_\infty$ The computational equation for the calculation of E is

$$(4.28) \qquad E_c \equiv 0.5\{(P_e^{-1}+E_3) + (Q_e^{-1}+E_4)\}+E_{51} .$$

Therefore

$$(4.29) \qquad \|E_5\|_\infty \le 0.5(\|E_3\|_\infty+\|E_4\|_\infty)+\|E_{51}\|_\infty .$$

4.4.3.1 <u>A Bound on $\|E_{51}\|_\infty$</u> Since E_{51} is the matrix of round-off errors incurred due to the addition of P_c^{-1} and Q_c^{-1}, and multiplication by .5 is done by shifting, it follows from (2.27) that

$$(4.30) \qquad \|E_{51}\|_\infty \leq 0.5\ \varepsilon(\|P_c^{-1}\|_\infty + \|Q_c^{-1}\|_\infty) .$$

The following theorem is a result of (4.17), (4.23), (4.24), (4.25), (4.26) and (4.27):

<u>Theorem 4.3</u> *Let* E_5 *denote the matrix of round-off errors in the calculation of* E *using Charmonman's Algorithm I (see Section 4.3), then*

$$(4.31) \qquad \|E_5\|_\infty \leq \tfrac{1}{2}\,\varepsilon\Bigg[\left\{1+\frac{g_P(2.005\ n^2+n^3)\|P_e^{-1}\|_\infty}{1-\varepsilon(\|A\|_\infty+\|B\|_\infty)\|P_e^{-1}\|_\infty}\right\}\|P_c^{-1}\|_\infty$$

$$+\left\{1+\frac{g_Q(2.005\ n^2+n^3)\|Q_e^{-1}\|_\infty}{1-2\varepsilon(\|A\|_\infty+2\|B\|_\infty)\|Q_e^{-1}\|_\infty}\right\}\|Q_c^{-1}\|_\infty$$

$$+\frac{(\|A\|_\infty+\|B\|_\infty)\|P_e^{-1}\|_\infty\|P_e^{-1}\|_\infty}{1-\varepsilon(\|A\|_\infty+\|B\|_\infty)\|P_e^{-1}\|_\infty}$$

$$+ \quad \left. \frac{2(\|A\|_\infty+2\|B\|_\infty)\|Q_e^{-1}\|_\infty\|Q_e^{-1}\|_\infty}{1-2\varepsilon(\|A\|_\infty+2\|B\|_\infty)\|Q_e^{-1}\|_\infty} \right] ,$$

provided

(4.32) $$\varepsilon(\|A\|_\infty+\|B\|_\infty)\|P_e^{-1}\|_\infty < 1 ,$$

and

(4.33) $$2\varepsilon(\|A\|_\infty+2\|B\|_\infty)\|Q_e^{-1}\|_\infty < 1 .$$

4.4.4 <u>A Bound on $\|E_6\|_\infty$</u> The computational equation for the calculation of F is

(4.34) $$F_c \equiv E+E_5-Q_e^{-1}+E_4 .$$

Therefore

(4.35) $$\|E_6\|_\infty \leq \|E_4\|_\infty+\|E_5\|_\infty .$$

The following theorem is a result of (4.25) and (4.31):

<u>Theorem 4.4</u> *If* E_6 *denotes the matrix of round-off errors incurred in the calculation of* F *using Charmonman's Algorithm I (see Section 4.3) subject to (4.32) and (4.33), then*

$$\|E_6\|_\infty \leq \|E_5\|_\infty + \varepsilon\{g_Q(2.005\ n^2+n^3)\|Q_c^{-1}\|_\infty + 2(\|A\|_\infty+2\|B\|_\infty)\}\ \frac{\|Q_e^{-1}\|_\infty}{1-2\varepsilon(\|A\|_\infty+2\|B\|_\infty)\|Q_e^{-1}\|_\infty} \quad . \tag{4.36}$$

4.4.5 <u>A Bound for Computed Inverse of R</u> If R_e^{-1} and R_c^{-1} denote the exact and computed inverse of a nonsingular block-symmetric matrix $R = \left[\begin{array}{c|c} A & B \\ \hline B & A \end{array}\right]$ of order $2n$, where A and B are of order n each, and if matrices $A+B$ and $A-B$ are nonsingular, we may write

$$R_e^{-1} = \left[\begin{array}{c|c} E & F \\ \hline F & E \end{array}\right] , \tag{4.37}$$

and

$$R_c^{-1} = \left[\begin{array}{c|c} E+E_5 & F+E_6 \\ \hline F+E_6 & E+E_5 \end{array}\right] , \tag{4.38}$$

where E_5 and E_6 denote the matrices of round-off errors incurred in the calculation of E and F, respectively. From (4.37) and (4.38) it follows that

$$\|R_c^{-1}-R_e^{-1}\|_\infty \leq \|E_5\|_\infty+\|E_6\|_\infty , \tag{4.39}$$

where E_5 and E_6 are bounded by (4.31) and (4.36) subject to the conditions (4.32) and (4.33).

4.5 Comparison of Upper Norm Bounds in Inversion

We now attempt to compare the upper norm bounds in (4.9) and (4.39). Since $\|\Delta_c^{-1}\|_\infty \le \|\Delta_e^{-1}\|_\infty + \|E_2\|_\infty$, and $\|\delta_c^{-1}\|_\infty \le \|\delta_e^{-1}\|_\infty + \|E_4\|_\infty$, it follows from (4.6) and (4.8) respectively, that

$$(4.40)\quad \|E_2\|_\infty \le \frac{\varepsilon\{g_\Delta(2.005\ n^2+n^3)+\|E_1\|_\infty\}\|\Delta_e^{-1}\|_\infty\|\Delta_e^{-1}\|_\infty}{1-\{\varepsilon g_\Delta(2.005\ n^2+n^3)+\|E_1\|_\infty\}\|\Delta_e^{-1}\|_\infty},$$

and

$$(4.41)\quad \|E_4\|_\infty \le \frac{\{\varepsilon g_\delta(2.005\ n^2+n^3)+\|E_3\|\}\|\delta_e^{-1}\|_\infty\|\delta_e^{-1}\|_\infty}{1-\{\varepsilon g_\delta(2.005\ n^2+n^3)+\|E_3\|_\infty\}\|\delta_e^{-1}\|_\infty},$$

where $\|E_1\|_\infty$ and $\|E_3\|_\infty$ are defined by (4.5) and (4.7), respectively. If computed inverse Δ_c^{-1} and δ_c^{-1} of Δ and δ, respectively are reasonable approximation to the true inverse, we may assume that

$$(4.42)\qquad \{\varepsilon g_\Delta(2.005\ n^2+n^3)+\|E_1\|_\infty\}\|\Delta_e^{-1}\|_\infty < 0.1\ ,$$

and

$$(4.43)\qquad \{\varepsilon g_\delta(2.005\ n^2+n^3)+\|E_3\|_\infty\}\|\delta_e^{-1}\|_\infty < 0.1\ .$$

It follows from (4.40) and (4.41) that

$$(4.44)\quad \|E_2\|_\infty \le 1.12\{\varepsilon g_\Delta(2.005\ n^2+n^3)+\|E_1\|_\infty\}\|\Delta_e^{-1}\|_\infty\|\Delta_e^{-1}\|_\infty\ ,$$

and

$$(4.45)\qquad \|E_4\|_\infty \le 1.12\{\varepsilon g_\delta(2.005\, n^2+n^3)+\|E_3\|_\infty\}\|\delta_e^{-1}\|_\infty\|\delta_e^{-1}\|_\infty .$$

Without loss of generality, we can assume that elements r_{ij} of matrix R have been scaled such that

$$(4.46)\qquad \max|r_{ij}| \le 1 .$$

Therefore

$$(4.47)\qquad \|A\|_\infty \le n ,$$

and

$$(4.48)\qquad \|B\|_\infty \le n .$$

Using (4.47), (4.48) in (4.5) and (4.7), respectively,

$$(4.49)\qquad \|E_1\|_\infty \le \varepsilon[n+n^2\{g_A(2.005\, n^2+n^3)\|A_e^{-1}\|_\infty+(2n+1)+n(n+2)\varepsilon\}$$

$$\times \|A_c^{-1}\|_\infty] ,$$

and

$$(4.50) \qquad \|E_3\|_\infty \le \varepsilon[n+n^2\{g_B(2.005\ n^2+n^3)\|B_e^{-1}\|_\infty+(2n+1)$$

$$+ n(n+2)\varepsilon\}\|B_c^{-1}\|_\infty] .$$

Therefore,

$$(4.51) \qquad \|E_1\|_\infty+\|E_3\|_\infty \le \varepsilon[2n+n^2\{(2.005\ n^2+n^3)$$

$$\times (g_A\|A_e^{-1}\|_\infty\|A_c^{-1}\|_\infty+g_B\|B_e^{-1}\|_\infty\|B_c^{-1}\|_\infty)$$

$$+ (2n+1)(\|A_c^{-1}\|_\infty+\|B_c^{-1}\|_\infty)\}] ,$$

with terms of order ε^2 ignored. In order to guarantee that the computed inverse of Δ and δ is a reasonable approximation to the true inverse, we assume that $\|\Delta_c^{-1}\|_\infty \le 1.01\|\Delta_e^{-1}\|_\infty$ and $\|\delta_c^{-1}\|_\infty \le 1.01\|\delta_e^{-1}\|_\infty$. Since $\|\Delta_e^{-1}\|_\infty \le \|R_e^{-1}\|_\infty$ and $\|\delta_e^{-1}\|_\infty \le \|R_e^{-1}\|_\infty$, we have from (4.9), (4.44), (4.45) and (4.51) that

$$(4.52) \quad \|R_c^{-1}-R_e^{-1}\|_\infty^{(S)} \leq 1.12\varepsilon[(g_\Delta+g_\delta)(2.005\ n^2+n^3)+2n$$

$$+ 1.01\{n^2(2.005\ n^2+n^3)(g_A\|A_e^{-1}\|_\infty+g_B\|B_e^{-1}\|_\infty)$$

$$+ (2n+1)(\|A_e^{-1}\|_\infty+\|B_e^{-1}\|_\infty)\}]\|R_e^{-1}\|_\infty\|R_e^{-1}\|_\infty ,$$

where $\|R_c^{-1}-R_e^{-1}\|_\infty^{(S)}$ denotes the ∞-norm of the matrix of round-off errors incurred in computing the inverse of a block-symmetric matrix R using Schur's Algorithm III (see Section 4.1).

Similarly, assuming that

$$(4.53) \qquad \varepsilon(\|A\|_\infty+\|B\|_\infty)\|P_e^{-1}\|_\infty < 0.1 ,$$

and

$$(4.54) \qquad 2\varepsilon(\|A\|_\infty+2\|B\|_\infty)\|Q_e^{-1}\|_\infty < 0.1 ,$$

it follows from (4.31), (4.36), (4.39), (4.47) and (4.48) that

$$(4.55)\qquad \|R_c^{-1}-R_e^{-1}\|_\infty^{(C)} \leq \varepsilon(\|P_c^{-1}\|_\infty+\|Q_c^{-1}\|_\infty)+1.12\varepsilon$$

$$\times \{(2.005\ n^2+n^3)(g_P\|P_c^{-1}\|_\infty\|P_e^{-1}\|_\infty+2g_Q\|Q_c^{-1}\|_\infty$$

$$\times \|Q_e^{-1}\|_\infty)+2n(\|P_e^{-1}\|_\infty^2+\|Q_e^{-1}\|_\infty^2)\}\ ,$$

where $\|R_c^{-1}-R_e^{-1}\|_\infty^{(C)}$ denotes the ∞-norm of the matrix of round-off errors incurred in computing the inverse of a block-symmetric matrix R using Charmonman's Algorithm I (see Section 4.3).

Assuming that $\|P_c^{-1}\|_\infty \leq 1.01\|P_e^{-1}\|_\infty$ and $\|Q_c^{-1}\|_\infty \leq 1.01\|Q_e^{-1}\|_\infty$, we have from (4.55),

$$(4.56)\qquad \|R_c^{-1}-R_e^{-1}\|_\infty^{(C)} \leq 1.01\ \varepsilon(\|P_e^{-1}\|_\infty+\|Q_e^{-1}\|_\infty)+1.12\varepsilon$$

$$\times \{1.01(2.005\ n^2+n^3)(g_P\|P_e^{-1}\|_\infty^2+2g_Q\|Q_e^{-1}\|_\infty^2\ )$$

$$+\ 2n(\|P_e^{-1}\|_\infty^2+\|Q_e^{-1}\|_\infty^2)\}\ .$$

Since $R_e^{-1} = \left[\begin{array}{c|c} E & F \\ \hline F & E \end{array}\right]$, it follows from the definition of ∞-norm that

$$(4.57) \qquad \|E\|_\infty \leq \|R_e^{-1}\|_\infty ,$$

and

$$(4.58) \qquad \|F\|_\infty \leq \|R_e^{-1}\|_\infty .$$

Further, since $P_e^{-1} = E+F$ and $Q_e^{-1} = E-F$, it follows from (4.57) and (4.58) that

$$(4.59) \qquad \|P_e^{-1}\|_\infty \leq 2\|R_e^{-1}\|_\infty ,$$

and

$$(4.60) \qquad \|Q_e^{-1}\|_\infty \leq 2\|R_e^{-1}\|_\infty .$$

Hence, from (4.55), (4.59) and (4.60) ,

$$(4.61) \quad \|R_c^{-1}-R_e^{-1}\|_\infty^{(C)} \leq \varepsilon[1.12\{4.04(2.005\ n^2+n^3)(g_P+2g_Q)$$

$$+ 32n\}+2.02]\|R_e^{-1}\|_\infty\|R_e^{-1}\|_\infty ,$$

provided

$$\|R_e^{-1}\|_\infty \geq 1 . \tag{4.62}$$

It is observed in practice that if the elements of a non-singular matrix are bounded by unity, then the elements of its inverse are generally greater than unity. Therefore in the light of (4.46), the assumption in (4.62) is generally true. It is thus obvious by comparison of (4.61) to (4.52) that the norm bound for the matrix of round-off errors incurred in computing the inverse of a block-symmetric matrix using Charmonman's Algorithm I (see Section 4.3) is considerably smaller than that using Schur's Algorithm III (see Section 4.1). In particular, let $g = \max(g_\Delta, g_\delta, g_Q, \|A_e^{-1}\|_\infty, \|B_e^{-1}\|_\infty)$, then for large n, the ratio of the upper norm bounds in (4.52) and (4.61) is

$$\frac{\|R_c^{-1}-R_e^{-1}\|_\infty^{(S)}}{\|R_c^{-1}-R_e^{-1}\|_\infty^{(C)}} \approx (gn)^2 . \tag{4.63}$$

Although, the actual errors depend on the nature of the elements of the matrices, it is reasonable and standard

practice to say that, in general, a method with smaller error bound is better than that with larger error bound.

4.6 Solution of System of Equations

The solution of the system of linear algebraic equations

$$Rx = b \ , \tag{4.64}$$

where the coefficient matrix R is block-symmetric, may be computed according to the algorithm given by Charmonman [1]. This algorithm requires smaller computer storage and smaller number of multiplications. It is based on the following theorem:

Theorem 4.5 *The solution of system of equations (4.64) which can be written in the form*

$$\left[\begin{array}{c|c} A & B \\ \hline B & A \end{array}\right] \left[\begin{array}{c} x_1 \\ \hline x_2 \end{array}\right] = \left[\begin{array}{c} b_1 \\ \hline b_2 \end{array}\right] , \tag{4.65}$$

is $x_1 = 0.5(y_1+y_2)$ *and* $x_2 = 0.5(y_1-y_2)$ *where* y_1 *and* y_2 *are solutions of* $(A+B)y_1 = b_1+b_2$ *and* $(A-B)y_2 = b_1-b_2$, *respectively.*

The theorem can be easily proved by equating terms on both sides of (4.65).

Charmonman's Algorithm II To compute the solution of (4.64):

Step 1: Compute $P=A+B$.

Step 2: Compute $Q=P-2B$.

Step 3: Compute $c_1=b_1+b_2$.

Step 4: Compute $c_2=c_1-2b_2$.

Step 5: Compute y_1 from $Py_1=c_1$.

Step 6: Compute y_2 from $Qy_2=c_2$.

Step 7: Compute $x_1=0.5(y_1+y_2)$.

Step 8: Compute $x_2=x_1-y_2$.

4.7 An Error Analysis of Solution of Equations

We now proceed to place norm bounds on the round-off error incurred in the computation of the solution of (4.64) using Charmonman's Algorithm II (see Section 4.6). We will use the same notations as in Section 3.2.

4.7.1 Bounds on $\|E_1\|_\infty$, $\|E_2\|_\infty$, $\|E_3\|_\infty$ and $\|E_4\|_\infty$

Upper norm bounds on the round-off errors incurred in the first four steps of Charmonman's Algorithm II follow from (2.27). Thus

$$\|E_1\|_\infty \le \varepsilon(\|A\|_\infty+\|B\|_\infty) \;, \tag{4.66}$$

$$\|E_2\|_\infty \le 2\varepsilon(\|A\|_\infty+2\|B\|_\infty) \;, \tag{4.67}$$

$$\|E_3\|_\infty \le \varepsilon(\|b_1\|_\infty+\|b_2\|_\infty) \;, \tag{4.68}$$

and

$$\|E_4\|_\infty \le 2\varepsilon(\|b_1\|_\infty+2\|b_2\|_\infty) \;. \tag{4.69}$$

4.7.2 <u>Bounds on $\|E_5\|_\infty$ and $\|E_6\|_\infty$</u> Let $y_e^{(1)}$ and $y_c^{(1)}$ denote the exact and computed solution of $Py_1=c_1$, then it can be easily shown that the computed solution is an exact solution of the perturbed system

$$(P+E_1+K)y_c^{(1)} = c_e^{(1)}+E_3 \;, \tag{4.70}$$

where $c_e^{(1)}$ denotes the exact value of c_1. A bound on K follows from (3.68).

We have

$$\|K\|_\infty \le \varepsilon g_P(2.005\; n^2+n^3) \;. \tag{4.71}$$

Let $h = y_c^{(1)} - y_e^{(1)}$, then from (4.70),

$$|h| = |P_e^{-1}\{E_3 - (E_1+K)y_c^{(1)}\}| , \tag{4.72}$$

or

$$|y_c^{(1)} - y_e^{(1)}| = |P_e^{-1}\{E_3 - (E_1+K)y_c^{(1)}\}| , \tag{4.73}$$

or

$$\|y_c^{(1)} - y_e^{(1)}\|_\infty \leq \|P_e^{-1}\|_\infty\{\|E_3\|_\infty + (\|E_1\|_\infty + \|K\|_\infty)\|y_c^{(1)}\|_\infty\} . \tag{4.74}$$

Using (4.66), (4.68) and (4.71) in (4.74) and noting that $E_5 = y_c^{(1)} - y_e^{(1)}$, we derive

$$\|E_5\|_\infty \leq \varepsilon\, [\, \{g_P(2.005\ n^2 + n^3) + \|A\|_\infty + \|B\|_\infty\}\|y_c^{(1)}\|_\infty + \|b_1\|_\infty + \|b_2\|_\infty]\|P_e^{-1}\|_\infty , \tag{4.75}$$

where g_P is the element of maximum modulus at all stages in the reduction of computed P by use of Gaussian elimination.

Similarly, the upper norm bound for the matrix of round-off errors incurred in computing the solution of the system $Qy_2=c_2$ is given by

$$(4.76)\qquad \|E_6\|_\infty \leq \varepsilon[\{g_Q(2.005\ n^2+n^3)+2(\|A\|_\infty+2\|B\|_\infty)\}\|y_c^{(2)}\|_\infty$$
$$+\ 2(\|b_1\|_\infty+2\|b_2\|_\infty)]\|Q_e^{-1}\|_\infty\ ,$$

where g_Q denotes the element of maximum magnitude in the transformation of computed Q using Gaussian elimination and $y_c^{(2)}$ denotes the computed solution of $Qy_2=c_2$.

4.7.3 <u>Bounds on $\|E_7\|_\infty$ and $\|E_8\|_\infty$</u> The computational equation for the calculation of x_1 is

$$(4.77)\qquad x_1 \equiv 0.5\{y_e^{(1)}+E_5+y_e^{(2)}+E_6\}+E_{71}\ .$$

Therefore

$$(4.78)\qquad |E_7| = |0.5(E_5+E_6)+E_{71}|\ ,$$

or

$$(4.79) \qquad \|E_7\|_\infty \le 0.5(\|E_5\|_\infty+\|E_6\|_\infty)+\|E_{71}\|_\infty .$$

Also,

$$(4.80) \qquad \|E_{71}\|_\infty \le 0.5(\|E_5\|_\infty+\|E_6\|_\infty) .$$

Hence, it follows from (4.75), (4.76), (4.79) and (4.80) that

$$(4.81) \qquad \|E_7\|_\infty \le \varepsilon[(\|b_1\|_\infty+\|b_2\|_\infty)\|P_e^{-1}\|_\infty+2(\|b_1\|_\infty+2\|b_2\|_\infty)\|Q_e^{-1}\|_\infty$$

$$+ \{g_P(2.005\ n^2+n^3)+\|A\|_\infty+\|B\|_\infty\}\|P_e^{-1}\|_\infty\|y_c^{(1)}\|_\infty$$

$$+ \{g_Q(2.005\ n^2+n^3)+2(\|A\|_\infty+2\|B\|_\infty)\}\|Q_e^{-1}\|_\infty\|y_c^{(2)}\|_\infty] .$$

Similarly, we have

$$(4.82) \qquad \|E_8\|_\infty \le \|E_6\|_\infty+\|E_7\|_\infty+\|E_{81}\|_\infty ,$$

where E_{81} is bounded by

$$(4.83) \qquad \|E_{81}\|_\infty \le \varepsilon\|x_c^{(1)}-y_c^{(2)}\|_\infty .$$

The result of this section is summarized in the following theorem:

Theorem 4.6 *If the solution of (4.64) is computed using Charmonman's Algorithm II (see Section 4.6), then the round-off error in the computed solution* x_c *is bounded by*

$$\|x_c - x_e\|_\infty \le \|E\|_6 + \|E_7\|_\infty + \varepsilon \|x_c^{(1)} - y_c^{(2)}\|_\infty , \tag{4.84}$$

where E_6 *and* E_7 *are defined by (4.76) and (4.81), respectively, and* x_e *denotes the exact solution of (4.64).*

4.8 Comparison of Upper Norm Bounds in System of Equations

We now proceed to compare the upper norm bounds on the round-off errors incurred in computing the solution of system of equations (4.64) using Charmonman's Algorithm II (see Section 4.6) and that obtained by using Schur's Algorithm II (see Section 3.5) for a block-symmetric coefficient matrix. Norm bound for the error in the computed x_1 and x_2 can be obtained from (3.66), (3.67), (3.70), (3.72), (3.84), (3.85) and (3.86) by replacing r by n, C by B and D by A.

Thus

$$(4.85)\quad \|x_c^{(1)}-x_e^{(1)}\|_\infty \leq \varepsilon[g_B(2.005\ n^2+n^3)\|x_c^{(1)}\|_\infty+\|b_2\|_\infty$$

$$+\ (n+1+n\varepsilon)\|x_c^{(2)}\|_\infty\|A\|_\infty]\|B_e^{-1}\|_\infty+\|x_c^{(2)}-x_e^{(2)}\|_\infty\|A\|_\infty\|B_e^{-1}\|_\infty\ ,$$

and

$$(4.86)\quad \|x_c^{(2)}-x_e^{(2)}\|_\infty \leq \varepsilon[\|b_2\|_\infty+\|A\|_\infty\|x_c^{(2)}\|_\infty+\{g_A(2.005\ n^2+n^3\|A_e^{-1}\|_\infty$$

$$+\ 1+2n+n(n+2)\varepsilon\}(\|b_1\|_\infty+\|B\|_\infty\|x_c^{(2)}\|_\infty)\|B\|_\infty\|A_c^{-1}\|_\infty$$

$$+\ g_\Delta(2.005\ n^2+n^3)\|x_c^{(2)}\|_\infty]\|\Delta_e^{-1}\|_\infty\ ,$$

where Δ_e^{-1} denotes the exact inverse of $(A-BA^{-1}B)$.

If we assume that

$$(4.87)\qquad \|B^{-1}\|_\infty\|A\|_\infty \geq 1\ ,$$

then the norm bound for the round-off error in computed x_c is given by

$$(4.88) \qquad \|x_c - x_e\|_\infty \le \varepsilon[g_B(2.005\, n^2+n^3)\|x_c^{(1)}\|_\infty + \|b_2\|_\infty$$

$$+ (n+1+n\varepsilon)\|x_c^{(2)}\|_\infty\|A\|_\infty]\|B_e^{-1}\|_\infty + \|x_c^{(2)} - x_e^{(2)}\|_\infty ,$$

where $\|x_c^{(2)} - x_e^{(2)}\|_\infty$ is bounded by (4.86).

Without loss of generality, we can assume that elements of the coefficient matrix R have been scaled such that

$$(4.89) \qquad |r_{ij}| \le 1 .$$

With this assumption, we certainly have

$$(4.90) \qquad \|A\|_\infty \le n ,$$

$$(4.91) \qquad \|B\|_\infty \le n .$$

Substituting (4.86) in (4.88) and using (4.90) and (4.91), it follows that the norm bound on round-off errors incurred in computing the solution of (4.64) by Schur's Algorithm II (see Section 3.5) is given by

$$(4.92) \quad \|x_c - x_e\|_\infty^{(S)} \le \varepsilon[g_B(2.005\ n^2+n^3)\|x_c^{(1)}\|_\infty + \|b_2\|_\infty$$

$$+ n(n+1+n\varepsilon)\|x_c^{(2)}\|_\infty]\|B_e^{-1}\|_\infty + \varepsilon n[\|b_2\|_\infty + n\|x_c^{(2)}\|_\infty$$

$$+ n\{g_A(2.005\ n^2+n^3)\|A_e^{-1}\|_\infty + 1 + 2n + n(n+2)\varepsilon\}$$

$$\times (\|b_1\|_\infty + n\|x_c^{(2)}\|_\infty)\|A_c^{-1}\|_\infty + g_\Delta(2.005\ n^2+n^3)\|x_c^{(2)}\|_\infty]$$

$$\times \|B_c^{-1}\|_\infty \|\Delta_e^{-1}\|_\infty \ .$$

Similarly, substituting (4.76) and (4.81) in (4.84) and using (4.90) and (4.91), we derive the following norm bound for the round-off errors incurred in computing the solution of (4.64) using Charmonman's Algorithm II (see Section 4.6):

$$(4.93) \quad \|x_c - x_e\|_\infty^{(C)} \le \varepsilon[\|x_c^{(1)}\|_\infty + \|y_c^{(2)}\|_\infty + (\|b_1\|_\infty + \|b_2\|_\infty)$$

$$\times \|P_e^{-1}\|_\infty + 2(\|b_1\|_\infty + 2\|b_2\|_\infty)\|Q_e^{-1}\|_\infty$$

$$+ \{2n + g_P(2.005\ n^2+n^3)\}\|P_e^{-1}\|_\infty \|y_c^{(1)}\|_\infty$$

$$+ 3\{6n + g_Q(2.005\ n^2+n^3)\}\|Q_e^{-1}\|_\infty \|y_c^{(2)}\|_\infty] \ .$$

It is thus obvious from (4.92) and (4.93) that the dominating term in the expression for round-off error incurred in computing the solution of (4.64) using Schur's Algorithm II (see Section 3.5) is $O(n^6)$, whereas the corresponding error term using Charmonman's Algorithm II (see Section 4.6) is $O(n^3)$. Wilkinson [17] has shown that the round-off error incurred in computing the solution of (4.64) using Gaussian elimination only, is also of order n^3. However, the actual errors depend on the nature of the elements of the coefficient matrices considered.

CHAPTER V

BAND MATRICES USING MINIMUM STORAGE

5.1 Solution of Equations

Gaussian elimination method (see Section 2.2) can be used to solve a system of linear equations with large band coefficient matrix. We will consider a general band matrix in which the number of co-diagonals on each side of the main diagonal are not equal. The concept of band matrix is useful only when the band width is appreciably smaller than the order of the matrix. The band width depends upon the ordering of the unknowns and the equations in the system. Evidently, row or column interchange will result in increase of band width and accordingly more storage space is required. Further, if complete pivoting is used the band structure may be completely lost. However, with partial pivoting, the number of upper co-diagonals may increase but the maximum increase can not be more than the number of lower co-diagonals. Also, the band characteristics of the given matrix are preserved. In order to take advantage of the band form of the given coefficient matrix, only the elements on the band are stored in the computer. One way to accomplish this is

to store the elements on the band row-wise. The LU decomposition of the given band matrix is then performed using an algorithm given in [7] based on Gaussian elimination, which uses minimum storage. Since partial pivoting is used the upper-triangular matrix U will involve as much storage as A in the worst case. Therefore, additional storage must be provided for the lower-triangular matrix L or if L is not saved, the right-hand sides must be processed simultaneously.

The algebraic basis of the decomposition of band matrix A into lower and upper triangular matrices is contained in the following theorem:

<u>Theorem 5.1</u> *If a square band matrix of order* n *with* p *lower and* q *upper co-diagonals has an* LU *decomposition, then* $L=(\ell_{ij})$ *and* $U=(u_{ij})$ *are triangular band matrices. That is,*

$$\ell_{ij} \neq 0 \quad \text{only for} \quad i=j,\ldots,j+p \ ,$$

$$u_{ij} \neq 0 \quad \text{only for} \quad j=i,\ldots,i+q \ .$$

The proof of this theorem follows from Theorem 2.1.

5.2 Upper Bound for the Pivot in the r^{th} Reduced Band Matrix

Let A denote an unsymmetric non-singular square band matrix of order n with p lower and q upper co-diagonals. Then using Gaussian forward elimination, the original matrix $A \equiv A^{(1)}$ is transformed successively into matrices $A^{(2)}, A^{(3)}, \ldots, A^{(r)}, \ldots, A^{(n)}$, such that each $A^{(r)}$ is equivalent to $A^{(1)}$ and the final reduced matrix $A^{(n)}$ is upper-triangular. We now proceed to place an upper bound on the magnitude of pivotal element in the r^{th} reduced matrix $A^{(r)}$ using partial pivoting. In the first few stages of reduction of A, the number of elements in the pivotal column is $p+1$ and the corresponding rows to be transformed are p in number excluding the pivotal row. These rows include one row which has not yet been modified. Thus from the growth of the elements in the pivotal sequence, it follows that every successive pivotal element, in the worst case, is the sum of preceding $p \leq n-1$ elements plus a, where $|a_{ij}^{(1)}| \leq a$. The undefined elements in the pivotal sequence are treated as zero.

Therefore,

$$|t_r| \leq |t_{r-p}| + |t_{r-(p-1)}| + \ldots + |t_{r-1}| + a \,, \tag{5.1}$$

where $t_r (r=p+1,p+2,\ldots,n)$ denote the pivot at the r^{th} transformation. Since each t_r is bounded by $2^{r-1}a$, we have from (5.1),

$$(5.2) \qquad |t_r| \leq \{2^{r-1}(1-2^{-p})+1\}a, \quad r=p+1,p+2\ldots,n, \quad p \leq n-1 .$$

The following theorem is a result of (5.2):

<u>Theorem 5.2</u> *Given a non-singular band matrix A of order n with p lower and q upper co-diagonals, then using Gaussian elimination with partial pivoting,*

$$(5.3) \qquad |a_{ij}^{(r)}| \leq \begin{cases} 2^{r-1}a, & r=1,2,\ldots,p, \\ \{2^{r-1}(1-2^{-p})+1\}a, & r=p+1,p+2\ldots,n, \\ & p \leq n-1, \end{cases}$$

where

$$(5.4) \qquad |a_{ij}^{(1)}| \leq a,$$

and $a_{ij}^{(k)}$ denotes the (i,j) element of the k^{th} reduced matrix $A^{(k)}$ with $A^{(1)}=A$.

5.3 An Error Analysis

In this section we place a norm bound on the round-off error incurred in the solution of a system of linear algebraic equations

$$(5.5) \qquad Ax = b \ ,$$

with non-singular band coefficient matrix of order n using Gaussian elimination with partial pivoting. We assume that A is unsymmetric with p lower and q upper co-diagonals. We may write the coefficient matrix as

$$
(5.6) \quad A = \begin{bmatrix}
a_{11} & a_{12} & \cdots & a_{1,q+1} & & & & \\
a_{21} & a_{22} & \cdots & a_{2,q+1} & a_{2,q+2} & & & \\
a_{31} & a_{32} & \cdots & a_{3,q+1} & a_{3,q+2} & a_{3,q+3} & & \\
\cdots & \cdots & \cdots & \cdots & \cdots & \cdots & & \\
a_{p+1,1} & a_{p+1,2} & \cdots & a_{p+1,q+1} & \cdots & & & \\
 & a_{p+2,2} & \cdots & a_{p+2,q+1} & \ddots & & & \\
 & & & \cdots & \cdots & \cdots & \cdots & \cdots \\
 & & & a_{n-1,n-p-1} & a_{n-1,n-p} & \cdots & a_{n-1,n} & \\
 & & & & a_{n,n-p} & \cdots & a_{nn} &
\end{bmatrix} .
$$

It has been shown (for example see Fox [5]) that the process of forward elimination with partial pivoting is equivalent to successive premultiplication of (5.6) by matrices $L_1R_1, L_2R_2, \ldots, L_{n-1}R_{n-1}$ where L_i are unit lower-triangular and R_i denotes the permutation matrix R_{ij} which accomplishes the partial pivoting at the i^{th} stage. Since the interchange of rows does not involve round-off error, we can assume, without loss of generality that the coefficient matrix has been reordered such that no interchange of rows is necessary. Then it can be easily shown that each L_i is of the form

$$(5.7)\quad L_i = \begin{bmatrix} 1 \\ 0 & 1 \\ 0 & 0 & 1 \\ \cdots & \cdots & \cdots \\ 0 & 0 & 0 & \ldots & 1 \\ 0 & 0 & 0 & \ldots & 0 & 1 \\ 0 & 0 & 0 & \ldots & 0 & m_{i+1,i} & 1 \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ 0 & 0 & 0 & \ldots & 0 & m_{p+i,i} & 0 & \ldots & 1 \\ 0 & 0 & 0 & \ldots & 0 & 0 & 0 & \ldots & 0 & \ldots & 1 \end{bmatrix}$$

for $i \leq n-p$, and

$$(5.8) \quad L_i = \begin{bmatrix} 1 \\ 0 & 1 \\ \cdots\cdots \\ 0 & 0 & \cdots & 1 \\ 0 & 0 & \cdots & 0 & 1 \\ 0 & 0 & \cdots & 0 & m_{i+1,i} & 1 \\ \cdots\cdots \\ 0 & 0 & \cdots & 0 & m_{n-1,i} & 0 & \cdots & 1 \\ 0 & 0 & \cdots & 0 & m_{n,i} & 0 & \cdots & 0 & 1 \end{bmatrix},$$

for $n-p < i \leq n-1$, where

$$(5.9) \qquad m_{ik} = \frac{a_{ik}^{(k)}}{a_{kk}^{(k)}}, \qquad \begin{array}{l} k = 1,\ldots,n-p, \\ i = k+1,\ldots,p+k, \end{array}$$

and

$$(5.10) \qquad m_{ik} = \frac{a_{ik}^{(k)}}{a_{kk}^{(k)}} \qquad \begin{array}{l} k = n-p+1,\ldots,n-1, \\ i = k+1,\ldots,n. \end{array}$$

Further, it can be shown that matrices $J=L_{n-1}L_{n-2}\dots L_1$ and $L=L_1^{-1}L_2^{-1}\dots L_{n-1}^{-1}$ are unit lower-triangular, and J and L have the form

$$(5.11)\ J = \begin{bmatrix} 1 & & & & & & & \\ m_{21} & 1 & & & & & & \\ m_{31} & m_{32} & 1 & & & & & \\ \cdots & \cdots & & & & & & \\ m_{p+1,1} & m_{p+1,2} & m_{p+1,3} & \cdots & & & & \\ 0 & m_{p+2,2} & m_{p+2,3} & \cdots & & & & \\ 0 & 0 & m_{p+3,3} & \cdots & & & & \\ \cdots & \cdots & \cdots & & & & & \\ 0 & 0 & \cdots & 0 & m_{n,n-p} & \cdots & m_{n,n-1} & 1 \end{bmatrix},$$

and

$$
(5.12)\quad L = \begin{bmatrix}
1 & & & & & & & & \\
-m_{21} & 1 & & & & & & & \\
-m_{31} & -m_{32} & 1 & & & & & & \\
\cdots\cdots & & & & & & & & \\
-m_{p+1,1} & -m_{p+1,2} & -m_{p+1,3} & \cdots & & & & & \\
0 & -m_{p+2,2} & -m_{p+2,3} & \cdots & & & & & \\
0 & 0 & -m_{p+3,3} & \cdots & & & & & \\
\cdots\cdots & & & & & & & & \\
0 & 0 & \cdots & 0 & -m_{n,n-p} & \cdots & -m_{n,n-1} & 1
\end{bmatrix}
$$

We have in fact performed a process equivalent to the decomposition of A into lower and upper triangular matrices, the upper-triangular matrix being the final reduced matrix $A^{(n)}$ which may have up to $p+q$ co-diagonals. We may therefore write

$$(5.13)\qquad A = LA^{(n)} ,$$

or

$$A = LU \quad , \tag{5.14}$$

in which L, the lower triangular matrix is given by (5.12). The equality sign in (5.14) can not be expected to be exactly satisfied due to round-off errors incurred in various computations performed. Wilkinson [16], has shown that computed L and U satisfy the perturbed equation

$$LU \equiv A+E \quad , \tag{5.15}$$

where E is the perturbation matrix. The solution of (5.5) is obtained by solving the triangular systems

$$Ly = b \quad , \tag{5.16}$$

and

$$Ux = y \quad . \tag{5.17}$$

The computational equations for computed y and x in (5.16) and (5.17) are given by

$$(L+\delta L)y \equiv b \quad , \tag{5.18}$$

and

$$(U+\delta U)x \equiv y \; . \tag{5.19}$$

Hence, computed x satisfies the computational equation

$$(L+\delta L)(U+\delta U)x \equiv b \; . \tag{5.20}$$

From (5.15),

$$(A+E+L\delta U+\delta LU+\delta L\delta U)x \equiv b \; . \tag{5.21}$$

If x_e and x_c denote the exact and computed solution of (5.5), then since $Ax_e=b$, it follows from (5.21) that

$$\|x_c-x_e\|_\infty \le \|A^{-1}\|_\infty(\|E\|_\infty+\|L\|_\infty\|\delta U\|_\infty+\|\delta L\|_\infty\|U\|_\infty + \|\delta L\|_\infty\|\delta U\|_\infty)\|x_c\|_\infty \; . \tag{5.22}$$

5.3.1 <u>A Bound for $\|E\|_\infty$</u> Using technqiues similar to those employed by Wilkinson [19] it can be easily shown that, in general for an unsymmetric band matrix A with p lower and q upper co-diagonals, the perturbation matrix $E = (e_{ij})$ is defined by,

$$
(5.23)\quad e_{ij} = \begin{cases} (i-)e, & i=2,\ldots,p+1, \text{ and } j=2,\ldots,q+1 \ ; \\ (i-j+q+1)e, & i=2,\ldots,p+1, \text{ and } j=q+2,\ldots,q+i \ ; \\ je, & i=2,\ldots,p+1, \text{ and } j=1,\ldots,i-1 \ ; \\ e_{p+1,j+p+1-i}, & i=p+2,\ldots,n-p-q, j=i-p,\ldots,p+q+i \ ; \\ e_{p+1,j+p+1-i}, & i=n-p-q+1,\ldots,n, \text{ and } j=i-p,\ldots,n; \\ 0, \text{ otherwise}, & \end{cases}
$$

where the element $re, (r=1,2,\ldots,p)$ represents the number of times the corresponding element in the matrix A was modified during the process of reduction of matrix A into upper-triangular matrix using Gaussian elimination,

and

$$(5.24)\qquad |re| < 3gr\varepsilon \quad , \quad j \geq i \ ,$$

and

$$(5.25)\qquad |re| < g(3r-2)\varepsilon \quad , \quad i > j \ .$$

g is defined as

$$g = \max_{i,j,k} |a_{ij}^{(k)}| \,. \tag{5.26}$$

A bound on g has already been given in Section 5.2. It can be easily seen that starting from the second row through $(p+1)^{th}$ row, the number of non-zero elements in the i^{th} row of the perturbation matrix E defined by (5.23) is $p+q+i-1$, $(i=2,\ldots,p+1)$; thereafter in each successive row this number decreases by 1. Thus $2p+q$ is the maximum number of non-zero elements in any row of (5.23). The result of this section is summarized in the following theorem:

<u>Theorem 5.3</u> *Given a non-singular band matrix A of order n with p lower and q upper co-diagonals. Then if $2p+q\le n$, the matrix E of round-off errors incurred in decomposing A into lower and upper triangular matrices using Gaussian elimination with partial pivoting is bounded by*

$$\|E\|_\infty \le \varepsilon g\{p(3p+3q+1)\}, \tag{5.27}$$

where g is defined by (5.26).

5.3.2 <u>Bounds on $\|L\|_\infty$ and $\|U\|_\infty$</u> Since partial pivoting ensures that the multipliers $|m_{ij}| \le 1$, therefore it immediately follows from (5.12) that

$$\|L\|_\infty \le (p+1) . \tag{5.28}$$

We now proceed to place an upper norm bound on the upper-triangular matrix U. The matrix U is of the form

$$U = \begin{bmatrix} a_{11}^{(1)} & a_{12}^{(1)} & \cdots & a_{1,p+q+1}^{(1)} & & & \\ & a_{22}^{(2)} & \cdots & a_{2,p+q+1}^{(2)} & a_{2,p+q+2}^{(2)} & & \\ & & \cdots & \cdots & \cdots & \cdots & \\ & & & & a_{n-1,n-1}^{(n-1)} & & a_{n-1,n}^{(n-1)} \\ & & & & & & a_{nn}^{(n)} \end{bmatrix} , \tag{5.29}$$

with p+q upper co-diagonals, in the worst case. Since g is the maximum element of any $|A^{(r)}|$, $(r=1,2,\ldots,n)$, therefore from (5.29),

$$\|U\|_\infty \le g(p+q+1) \ . \tag{5.30}$$

5.3.3 Bounds for $\|\delta L\|_\infty$ and $\|\delta U\|_\infty$ Error analysis of the solution of triangular sets of equations (5.16) and (5.17) is analogous to that given by Wilkinson [17]. Using a technique similar to that employed by Wilkinson [17], it follows using (2.18), (2.19) and (2.20) that computed solution of (5.16) is an exact solution of $(L+\delta L)x=b$, where δL is bounded by

$$(5.31)\quad |\delta L| \le |\varepsilon| \begin{bmatrix} 0 & & & & & & \\ 2|m_{21}| & 1 & & & & & \\ 3|m_{31}| & 2|m_{32}| & 1 & & & & \\ \cdots\cdots & \cdots\cdots & & & & & \\ (p+1)|m_{p+1,1}| & p|m_{p+1,2}| & (p-1)|m_{p+1,3}| & \cdots & 2|m_{p+1,p}| & 1 & \\ & \cdots\cdots & \cdots\cdots & \cdots & \cdots\cdots & & \\ & & (p+1)|m_{n,n-p}| & \cdots & 2|m_{n,n-1}| & & 1 \end{bmatrix}$$

Since the elements m_{ij} of lower-triangular matrix L are bounded by unity, therefore

$$(5.32) \qquad \|\delta L\|_\infty \le \tfrac{1}{2}\varepsilon(p+1)(p+2) .$$

Similarly, it can be shown that the computed solution of (5.17) is an exact solution of $(U+\delta U)x=y$, where δU is bounded by

$$(5.33)\quad |\delta U| \leq |\varepsilon| \begin{bmatrix} 2|u_{11}| & c|u_{12}| & c|u_{13}| & (c-1)|u_{14}| & \cdots & 2|u_{1,c+1}| & & & & \\ & & & \cdots\cdots\cdots & & & & & & \\ & & 2|u_{d-1,d-1}| & c|u_{d-1,d}| & (c-1)|u_{d-1,d+1}| & \cdots & 2|u_{d-1,n}| & & & \\ & & & & \cdots\cdots\cdots & & & & & \\ & & & & & & & 2|u_{n-2,n-2}| & 2|u_{n-2,n-1}| & 2|u_{n-2,n}| \\ & & & & & & & & 2|u_{n-1,n-1}| & |u_{n-1,n}| \\ & & & & & & & & & |u_{nn}| \end{bmatrix},$$

where $$c = p+q, \; d = n-c \; ,$$

and

$$U = A^{(n)} \; .$$

Hence,

$$\|\delta U\|_\infty \leq \varepsilon \tfrac{1}{2}(p+q+1)(p+q+2) \; . \tag{5.34}$$

The following theorem is a result of (5.22), (5.27), (5.28), (5.30), (5.32) and (5.34):

Theorem 5.4 *Let* x_e *and* x_c *denote the exact and computed solution of the system of linear algebraic equations,*

$$Ax = b \tag{5.35}$$

where A *is a band coefficient matrix with* p *lower and* q *upper co-diagonals. Then, if the terms of order* ε^2 *are neglected, the round-off error in computing the solution of (5.35) using Gaussian elimination with*

partial pivoting is bounded by

$$(5.36) \qquad \|x_c - x_e\|_\infty \le \tfrac{1}{2}\varepsilon g\{2p(3p+3q+1)+(p+1)(p+q+1)$$

$$\times\ (2p+q+4)\}\|A_e^{-1}\|_\infty \|x_c\|_\infty ,$$

for $\qquad 2p+q \le n$.

5.4 Inversion

Inversion of band matrices using minimum storage can be carried out by solving (5.5) with right-hand side replaced by an identity matrix of order n. The solution matrix is stored in the successive locations of right-hand side matrix, therefore no extra storage is necessary for the solution matrix.

5.5 Error Analysis of Inversion

The error analysis of inversion of a band matrix is analogous to that given in the last section. Using the results of Section 5.3 it can be shown that,

$$(5.37) \qquad \|A_c^{-1} - A_e^{-1}\|_\infty \le \tfrac{1}{2}\varepsilon g\{2p(3p+3q+1)+(p+1)(p+q+1)$$

$$\times \ (2p+q+4)\}\|A_e^{-1}\|_\infty \|A_c^{-1}\|_\infty \ ,$$

where A_e^{-1} and A_c^{-1} denote the exact and computed inverse of A, respectively. g is the element of maximum magnitude at all stages in the decomposition of A into lower and upper-triangular matrices.

CHAPTER VI

NUMERICAL RESULTS

Numerical experiments were carried out on IBM System 360/67 computer at the University of Alberta. FORTRAN IV source language was used for general and block-symmetric matrices, and APL\360 for band matrices. Several matrices with known exact inverses were used in testing the algorithms for inversion and solution of linear algebraic equations. The right-hand sides of the systems of equations were chosen arbitrarily. The exact solution of the system of equations $Ax=b$ was obtained by computing the product $A_e^{-1}b$, where A_e^{-1} is the exact inverse of the coefficient matrix A, and b the vector of right-hand sides. Norm bounds for the round-off errors as given in Chapters III, IV and V were computed in the above computer programs. A comparison is made between the predicted and actual relative round-off errors.

6.1 Test Data

The following test matrices were used for inversion and solution of system of equations. The matrices denoted by A_1, A_2, A_3, A_4, A_5 are due to Newman and Todd [10], while matrices A_6 and A_7 are due to Charmonman and Julius [2].

The matrix A_1 is defined by

$$(6.1) \qquad a_{ij} = \left[\frac{2}{n+1}\right]^{1/2} \sin\left[\frac{ij\pi}{n+1}\right] .$$

Since A_1 is symmetric and orthogonal, we have

$$(6.2) \qquad A_1^{-1} = A_1 .$$

The matrix A_2 is defined by

$$(6.3) \qquad a_{ij} = \begin{cases} i/j , & i \le j ; \\ j/i , & i > j . \end{cases}$$

The inverse of A_2 is symmetric, triple-diagonal matrix with

$$(6.4) \qquad a_{ij} = \begin{cases} \dfrac{4i^3}{4i^2-1} , & i=j, \quad i < n ; \\[2ex] \dfrac{n^2}{2n-1} , & i=j=n; \\[2ex] \dfrac{-i(i+1)}{2i+1} , & i < j, \ |i-j|=1 ; \\[2ex] a_{ji} , & i > j, \ |i-j|=1 ; \\[2ex] 0 , & |i-j| > 1 . \end{cases}$$

The matrix A_3 is defined by

$$(6.5) \qquad a_{ij} = 2\min(i,j)-1 ,$$

and its inverse is a triple-diagonal matrix with

$$(6.6) \qquad a_{ij} = \begin{cases} 1.5 , & i=j=1 ; \\ 0.5 , & i=j=n ; \\ 1 , & 1<i=j<n ; \\ -0.5 , & |i-j|=1 ; \\ 0 , & |i-j|>1 . \end{cases}$$

The matrix A_4 is defined by

$$(6.7) \qquad a_{ij} = \begin{cases} -2 , & i=j ; \\ 1 , & |i-j|=1 ; \\ 0 , & |i-j|>1 . \end{cases}$$

Its inverse is given by

$$(6.8) \qquad a_{ij} = \begin{cases} \dfrac{-i(n-j+1)}{n+1} , & i \le j ; \\ a_{ji} , & i > j . \end{cases}$$

The matrix A_5 is defined by

$$(6.9) \qquad a_{ij} = \begin{cases} 5, & i=j=1,\ i=j=n ; \\ 6, & i=j ; \\ -4, & |i-j|=1 ; \\ 1, & |i-j|=2 , \\ 0, & \text{otherwise} . \end{cases}$$

It can be noted that

$$(6.10) \qquad A_5 = A_4^2 ,$$

where A_4 is defined by (6.7). Thus

$$(6.11) \qquad A_5^{-1} = (A_4^{-1})^2 ,$$

where A_4^{-1} is defined by (6.8).

Matrices A_6 and A_7 have been constructed using the following definition due to Charmonman and Julius [2].

<u>Definition 6.1</u> *An r-circulant is a square matrix of order n in which the i^{th} row, $i=2,3,\ldots,n$, is obtained from the $(i-1)^{th}$ row by cyclically shifting each element r places to the right.*

The word "row" may be replaced by "column" if the word "right" is replaced by "down".

The matrices A_6 and A_7 are r-circulants with first rows $\{a, a+h, \ldots, a+(n-1)h\}$ and $\{a, ah, \ldots, ah^{n-1}\}$, respectively. The inverse of A_6 is the s-circulant with the first column $\{b-\alpha, b, \ldots, b, b+\alpha\}^T$, where s satisfies

$$(6.12) \qquad rs = kn+1 ,$$

for some integer k, and

$$(6.13) \qquad b = \frac{2}{n^2\{2a+(n-1)h\}} ,$$

and

$$(6.14) \qquad \alpha = \frac{1}{n\ h} .$$

The inverse of A_7 is also an s-circulant, where s is defined as in (6.12) with first column $\{b, 0, \ldots, 0, -hb\}^T$,

where

$$(6.15) \qquad b = \frac{1}{a(1-h^n)} .$$

6.2 Summary of the Results

6.2.1 General Matrices Matrices A_1, A_2, and A_3 of Section 6.1 were used in testing the algorithms given in Chapter III for inversion of general matrices and solution of system of equations with general coefficient matrices.

Table 6.1 gives the predicted relative errors and the actual relative errors incurred in the computation of inverses of general matrices using Schur's Algorithm I (see Section 3.1) and Gaussian elimination with complete pivoting. The predicted absolute error $\|A_c^{-1}-A_e^{-1}\|_\infty$ using the above algorithms is given by (3.60) and (3.17), respectively, and the predicted relative error is defined as

$$(6.16) \qquad \frac{\|A_c^{-1}-A_e^{-1}\|_\infty}{\|A_e^{-1}\|_\infty} .$$

Since the exact inverse of matrix A is known, actual relative error can be easily computed.

Table 6.2 gives the predicted relative errors and the actual relative errors incurred in the computation of solution of systems of equations using Schur's Algorithm II (see Section 3.5) and Gaussian elimination with complete pivoting. The predicted absolute error $\|x_c-x_e\|_\infty$ using

these algorithms is given by (3.77) and/or (3.87), and (3.17) with A_c^{-1} replaced by x_c, respectively. The predicted relative error is defined to be

$$\frac{\|x_c - x_e\|_\infty}{\|x_e\|_\infty} \tag{6.17}$$

The exact solution can be computed since the exact inverse of the coefficient matrix is known. As a result, therefore, the actual relative error can also be easily computed.

6.2.2 Block-Symmetric Matrices Matrices A_6 and A_7 of Section 6.1 were used in testing the algorithms given in Chapter IV for inversion of block-symmetric matrices and solution of system of equations with block-symmetric coefficient matrices. The three parenthesized numbers after the matrix A_6 and A_7 are, respectively, the values of a, h, and r used to construct these matrices.

Table 6.3 gives the predicted relative errors and actual relative errors incurred in the computation of inverses of block-symmetric matrices using Schur's Algorithm III (see Section 4.1), Charmonman's Algorithm I (see Section 4.4) and Gaussian elimination with complete pivoting. The predicted absolute error $\|A_c^{-1} - A_e^{-1}\|_\infty$ using the above algorithms is given by (4.9), (4.39) and (3.17) respectively, and the predicted relative error is defined by (6.16). The actual relative error can be computed since the exact inverse is known.

Table 6.4 gives the predicted relative errors and the actual relative errors incurred in the computation of solution of systems of equations with block-symmetric coefficient matrices using Schur's Algorithm II (see Section 3.5), Charmonman's Algorithm II (see Section 4.6), and Gaussian elimination with complete pivoting. The predicted absolute error $\|x_c - x_e\|_\infty$ using these algorithms is given by (3.87), (4.84), and (3.17) with A_c^{-1} replaced by x_c respectively, and the predicted relative error is given by (6.17). The exact solution can be computed since the exact inverse of the coefficient matrix is known.

6.2.3 Band Matrices Matrices A_4 and A_5 of Section 6.1 were used in testing the algorithm given in Chapter V for inversion of band matrices and solution of system of equations with band coefficient matrices.

Table 6.5 gives the predicted relative errors and the actual relative error incurred in the computation of inverses of band matrices using Gaussian elimination with partial pivoting with and without minimum storage. The predicted absolute error $\|A_c^{-1} - A_e^{-1}\|_\infty$ is given by (5.37) and (3.17) respectively, and the predicted relative error is given by (6.16). The actual relative error is computed using the given exact inverse.

Table 6.6 gives the predicted relative errors and actual relative error incurred in computing solution of systems of equations with band coefficient matrices using Gaussian elimination with partial pivoting with and without minimum storage. The predicted absolute error $\|x_c-x_e\|_\infty$ is given by (5.36), and (3.17) with A_c^{-1} replaced by x_c, respectively, and the predicted relative error is given by (6.17). The exact solution and therefore the actual relative error can be computed since the exact inverse of the coefficient matrix is known.

TABLE 6.1

GENERAL MATRICES-RELATIVE ERROR

IN INVERSION

Matrix	Order	Schur's Algorithm I		Gaussian elimination	
		Predicted	Actual	Predicted	Actual
A_1	5	0.1633E+03	0.4730E-05	0.3116E-03	0.8022E-06
A_2	5	0.1158E+04	0.1503E-05	0.6680E-03	0.4920E-06
A_3	5	0.7690E+03	0.2176E-05	0.1503E-02	0.9970E-06
A_1	10	0.1892E+06	0.9018E-02	0.4022E-02	0.7940E-05
A_2	10	0.6921E+04	0.4385E-05	0.1030E-01	0.2217E-05
A_3	10	0.4564E+06	0.7764E-05	0.2175E-01	0.3412E-05
A_1	11	0.3244E+06	0.8103E-01	0.5700E-02	0.1049E-04
A_2	11	0.6922E+04	0.4314E-05	0.1501E-01	0.2691E-05
A_3	11	0.4564E+06	0.1016E-04	0.3151E-01	0.1555E-04
A_1	12	0.6150E+06	0.3881E-01	0.7978E-02	0.9466E-05
A_2	12	0.2790E+05	0.2287E-05	0.2116E-01	0.3332E-05
A_3	12	0.1576E+06	0.9605E-05	0.4423E-01	0.7734E-05
A_1	13	0.1415E+07	0.5737	0.1101E-01	0.9613E-05
A_2	13	0.8641E+05	0.6597E-05	0.2902E-01	0.2588E-05
A_3	13	0.4564E+06	0.7130E-05	0.6046E-01	0.7515E-05
A_1	14	0.1999E+07	0.2713	0.1431E-01	0.1076E-04
A_2	14	0.8641E+05	0.6073E-05	0.3889E-01	0.3457E-05
A_3	14	0.4564E+06	0.8556E-05	0.8078E-01	0.9014E-05

TABLE 6.1 (Continued)

Matrix	Order	Schur's Algorithm I		Gaussian elimination	
		Predicted	Actual	Predicted	Actual
A_1	15	0.2850E+07	0.3168E+01	0.1852E-01	0.4934E-05
A_2	15	0.8642E+05	0.6801E-05	0.5108E-01	0.4577E-05
A_3	15	0.1124E+07	0.7053E-05	0.1058	0.1699E-04
A_1	16	0.4474E+07	0.1418E+02	0.2466E-01	0.1026E-04
A_2	16	0.2223E+06	0.1174E-04	0.6594E-01	0.4782E-05
A_3	16	0.1124E+07	0.1035E-04	0.1363	0.2448E-04
A_1	17	0.6870E+07	0.1191E+02	0.3040E-01	0.8671E-05
A_2	17	0.5020E+06	0.1421E-04	0.8381E-01	0.8317E-05
A_3	17	0.1124E+07	0.1131E-04	0.1730	0.2553E-04
A_1	18	0.1006E+08	0.1096E+03	0.3743E-01	0.8416E-05
A_2	18	0.5020E+06	0.1559E-04	0.1051	0.7826E-05
A_3	18	0.2459E+07	0.9578E-05	0.2164	0.2511E-04
A_1	19	0.1428E+08	0.4851E+03	0.4652E-01	0.8435E-05
A_2	19	0.1027E+07	0.1628E-04	0.1302	0.8280E-05
A_3	19	0.4910E+07	0.2604E-04	0.2676	0.1855E-04
A_1	20	0.1970E+08	0.5608E+03	0.5745E-01	0.1006E-04
A_2	20	0.1027E+07	0.2249E-04	0.1595	0.5574E-05
A_3	20	0.4910E+07	0.3071E-04	0.3274	0.4432E-04

TABLE 6.2

GENERAL MATRICES-RELATIVE

ERROR IN SOLUTION OF EQUATIONS

Coefficient Matrix	Order	Right-hand Side	Schur's Algorithm II		Gaussian Elimination	
			Predicted	Actual	Predicted	Actual
A_1	5	Vector of 2s'	0.1563E-02	0.1770E-05	0.3116E-03	0.4426E-06
A_2	5	"	0.1103E-01	0.2816E-05	0.6680E-03	0.3576E-05
A_3	5	"	0.8310E-02	0.1431E-05	0.1503E-02	0.9536E-06
A_1	10	"	0.1806E+01	0.7520E-04	0.4022E-02	0.4150E-05
A_2	10	"	0.2988E+01	0.8091E-05	0.1030E-01	0.6437E-05
A_3	10	"	0.4362	0.9537E-06	0.2175E-01	0.1907E-05
A_1	11	"	0.6517E+01	0.4529E-03	0.5699E-02	0.2873E-05
A_2	11	"	0.6271E+01	0.1314E-04	0.1501E-01	0.4291E-05
A_3	11	"	0.5533	0.9537E-06	0.3151E-01	0.9537E-06
A_1	12	"	0.1788E+02	0.3085E-03	0.7978E-02	0.2696E-05
A_2	12	"	0.9538E+01	0.1261E-04	0.2116E-01	0.4828E-05
A_3	12	"	0.1071E+01	0.1431E-05	0.4424E-01	0.9537E-06

TABLE 6.2 (continued)

Coefficient Matrix	Order	Right-hand Side	Schur's Algorithm II		Gaussian Elimination	
			Predicted	Actual	Predicted	Actual
A_1	13	"	0.4652E+02	0.1991E-03	0.1101E-01	0.4344E-05
A_2	13	"	0.1393E+02	0.2831E-04	0.2902E-01	0.1252E-04
A_3	13	"	0.1656E+01	0.9537E-06	0.6046E-01	0.1907E-05
A_1	14	"	0.1354E+02	0.7088E-03	0.1431E-01	0.3314E-05
A_2	14	"	0.2569E+02	0.2966E-04	0.3889E-01	0.1099E-04
A_3	14	"	0.2595E+01	0.1907E-05	0.8077E-01	0.1907E-05
A_1	15	"	0.2594E+03	0.2024E-02	0.1852E-01	0.1939E-05
A_2	15	"	0.4465E+02	0.2132E-04	0.5108E-01	0.1139E-04
A_3	15	"	0.3766E+01	0.6386E-06	0.1058	0.9537E-06
A_1	16	"	0.8214E+03	0.3203E-02	0.2466E-01	0.3606E-05
A_2	16	"	0.6102E+02	0.2825E-04	0.6594E-01	0.9522E-05
A_3	16	"	0.5604E+01	0.1043E-05	0.1363	0.9537E-06
A_1	17	"	0.1710E+04	0.1415E-02	0.3039E-01	0.2421E-05
A_2	17	"	0.8145E+02	0.3742E-04	0.8381E-01	0.1556E-04
A_3	17	"	0.8032E+01	0.9838E-06	0.1728	0.9537E-06

TABLE 6.2 (continued)

Coefficient Matrix	Order	Right-hand Side	Schur's Algorithm II		Gaussian Elimination	
			Predicted	Actual	Predicted	Actual
A_1	18	"	0.3014E+04	0.1950E-02	0.3743E-01	0.2268E-05
A_2	18	"	0.1316E+03	0.3589E-04	0.1051	0.1430E-04
A_3	18	"	0.1107E+02	0.1407E-05	0.2163	0.1907E-05
A_1	19	"	0.4367E+04	0.4363E-02	0.4651E-01	0.2388E-05
A_2	19	"	0.1703E+03	0.3062E-04	0.1301	0.1135E-04
A_3	19	"	0.1477E+02	0.1177E-05	0.2676	0.1430E-05
A_1	20	"	0.1488E+05	0.2595E-01	0.5744E-01	0.5015E-05
A_2	20	"	0.3085E+03	0.3639E-04	0.1595	0.1421E-04
A_3	20	"	0.2072E+02	0.2086E-05	0.3273	0.2384E-05

TABLE 6.3

BLOCK-SYMMETRIC MATRICES-RELATIVE

ERROR IN INVERSION

Matrix	Order	Schur's Algorithm III		Charmonman's Algorithm I		Gaussian Elim.	
		Predicted	Actual	Predicted	Actual	Predicted	Actual
$A_6(0.5,2,1)$	4	0.1001E-02	0.5085E-06	0.9632E-04	0.6232E-06	0.9163E-04	0.5346E-06
$A_6(0.2\ 5,1,1)$	4	0.1001E-02	0.6388E-06	0.9632E-04	0.3129E-06	0.9163E-04	0.4172E-06
$A_6(0.2\ 5,2,1)$	4	0.8658E-03	0.8911E-06	0.9440E-04	0.2066E-06	0.9163E-04	0.2712E-06
$A_6(1.5,2,1)$	4	0.1784E-02	0.4559E-06	0.1039E-03	0.5364E-06	0.9545E-04	0.6571E-06
$A_6(1.5,3,1)$	4	0.1340E-02	0.1709E-05	0.1002E-03	0.3178E-06	0.9163E-04	0.4569E-06
$A_6(1.5,4,1)$	4	0.1158E-02	0.9993E-06	0.9824E-04	0.3681E-06	0.9163E-04	0.2103E-06
$A_6(0.5,3.5,1)$	4	0.8837E-03	0.1539E-05	0.9468E-04	0.3621E-06	0.9163E-04	0.3621E-06
$A_6(1,1,1)$	6	-	-	0.3077E-03	0.5918E-06	0.3010E-03	0.8673E-06
$A_7(1,1,2)$	6	-	-	0.2035E-03	0.3129E-06	0.2094E-03	0.0
$A_6(1,1,1)$	8	-	-	0.6576E-03	0.7978E-06	0.6615E-03	0.7686E-06
$A_7(1,1,2)$	8	-	-	0.4224E-03	0.0	0.4598E-03	0.0
$A_6(1,1,1)$	10	-	-	0.1197E-02	0.1037E-05	0.1228E-02	0.8161E-06
$A_7(1,1,2)$	10	-	-	0.7606E-03	0.1985E-06	0.8595E-03	0.0

TABLE 6.3 (continued)

Matrix	Order	Schur's Algorithm III		Charmonman's Algorithm I		Gaussian Elimination	
		Predicted	Actual	Predicted	Actual	Predicted	Actual
$A_6(1,1,1)$	12	-	-	0.1967E-02	0.1126E-05	0.2047E-02	0.1449E-05
$A_7(1,1,2)$	12	-	-	0.1224E-02	0.0	0.1443E-02	0.0
$A_6(1,1,1)$	14	-	-	0.3007E-02	0.2593E-05	0.3163E-02	0.1833E-05
$A_7(1,1,2)$	14	-	-	0.1899E-02	0.3179E-06	0.2244E-02	0.0
$A_6(1,1,1)$	16	-	-	0.4357E-02	0.1959E-05	0.4622E-02	0.1050E-05
$A_7(1,1,2)$	16	-	-	0.2752E-02	0.7947E-06	0.3297E-02	0.0
$A_6(1,1,1)$	18	-	-	0.6056E-02	0.3447E-05	0.6471E-02	0.9925E-05
$A_7(1,1,2)$	18	-	-	0.3829E-02	0.1987E-06	0.4636E-02	0.0
$A_6(1,1,1)$	20	-	-	0.8146E-02	0.5503E-05	0.8754E-02	0.1283E-04
$A_7(1,1,2)$	20	-	-	0.5155E-02	0.7948E-06	0.6296E-02	0.0

TABLE 6.4

BLOCK-SYMMETRIC MATRICES—RELATIVE ERROR

IN SOLUTION OF EQUATIONS

Coefficient Matrix	Order	Right-hand Side	Schur's Algorithm II		Charmonman's Algorithm II		Gaussian Elimination	
			Predicted	Actual	Predicted	Actual	Predicted	Actual
$A_6(0\cdot5,2,1)$	4	Vector of 2s'	0.3226E-03	0.5424E-05	0.9649E-04	0.8345E-06	0.9163E-04	0.1252E-05
$A_6(0\cdot2\ 5,1,1)$	4	"	0.3226E-03	0.3129E-05	0.9648E-04	0.2086E-06	0.9163E-04	0.1252E-05
$A_6(0\cdot2\ 5,2,1)$	4	"	0.3022E-03	0.1201E-04	0.9190E-04	0.7748E-06	0.9163E-04	0.2325E-05
$A_6(1\cdot5,2,1)$	4	"	0.4391E-03	0.1126E-04	0.1150E-03	0.1609E-05	0.9544E-04	0.3218E-05
$A_6(1\cdot5,3,1)$	4	"	0.3744E-03	0.1645E-04	0.1057E-03	0.7152E-06	0.9163E-04	0.2861E-05
$A_6(1\cdot5,4,1)$	4	"	0.3468E-03	0.5364E-05	0.1011E-03	0.8941E-06	0.9163E-04	0.2682E-05
$A_6(0\cdot5,3\cdot5,1)$	4	"	0.3049E-03	0.1371E-05	0.9255E-04	0.0	0.9163E-04	0.2056E-05
$A_6(1,1,1)$	6	"	-	-	0.3162E-03	0.1251E-05	0.3010E-03	0.2503E-05
$A_7(1,1,2)$	6	"	-	-	0.1630E-03	0.0	0.2094E-03	0.0
$A_6(1,1,1)$	8	"	-	-	0.6394E-03	0.3219E-05	0.6615E-03	0.6437E-05
$A_7(1,1,2)$	8	"	-	-	0.3098E-03	0.0	0.4598E-03	0.0
$A_6(1,1,1)$	10	"	-	-	0.1129E-02	0.2459E-05	0.1228E-02	0.5224E-05
$A_7(1,1,2)$	10	"	-	-	0.5343E-03	0.0	0.8595E-03	0.0

TABLE 6.4 (continued)

Coefficient Matrix	Order	Right-Hand Side	Schur's Algorithm II		Charmonman's Algorithm II		Gaussian Elimination	
			Predicted	Actual	Predicted	Actual	Predicted	Actual
$A_6(1,1,1)$	12	"	-	-	0.1818E-02	0.1903E-04	0.2047E-02	0.2935E-04
$A_7(1,1,2)$	12	"	-	-	0.8550E-03	0.0	0.1443E-02	0.0
$A_6(1,1,1)$	14	"	-	-	0.2742E-02	0.9974E-05	0.3163E-02	0.1350E-04
$A_7(1,1,2)$	14	"	-	-	0.1290E-02	0.0	0.2244E-02	0.0
$A_6(1,1,1)$	16	"	-	-	0.3935E-02	0.1064E-04	0.4622E-02	0.1216E-04
$A_7(1,1,2)$	16	"	-	-	0.1857E-02	0.0	0.3240E-02	0.0
$A_6(1,1,1)$	18	"	-	-	0.5432E-02	0.3281E-04	0.6471E-02	0.5255E-04
$A_7(1,1,2)$	18	"	-	-	0.2573E-02	0.0	0.4636E-02	0.0
$A_6(1,1,1)$	20	"	-	-	0.7266E-02	0.1800E-04	0.8754E-02	0.1995E-04
$A_7(1,1,2)$	20	"	-	-	0.3456E-02	0.0	0.6296E-02	0.0

TABLE 6.5

BAND MATRICES-RELATIVE ERROR

IN INVERSION

Matrix	Order	Gaussian elimination with minimum storage	Gaussian elimination	Actual
		Predicted		
A_4	5	1.6786E-14	1.0499E-13	2.6923E-15
A_5	5	4.2002E-13	7.2827E-13	7.3275E-14
A_4	10	5.1292E-14	2.1992E-12	3.4681E-14
A_5	10	5.6658E-12	6.7345E-11	1.9081E-12
A_4	11	6.1042E-14	3.4306E-12	5.0113E-14
A_5	11	8.2035E-12	1.2781E-10	3.9853E-12
A_4	12	7.0721E-14	5.0937E-12	6.8709E-14
A_5	12	1.3292E-11	2.2620E-10	7.6360E-12
A_4	13	8.2019E-14	7.4282E-12	6.2204E-14
A_5	13	1.5469E-11	3.8841E-9	1.2327E-11
A_4	14	9.3259E-14	1.0448E-11	1.7761E-14
A_5	14	2.0234E-11	6.3404E-10	3.0414E-11
A_4	15	1.0611E-13	1.4499E-11	1.5174E-14
A_5	15	2.6719E-11	1.0122E-9	3.9568E-12
A_4	16	1.1890E-13	1.9574E-11	1.8581E-14
A_5	16	3.4077E-11	1.5552E-9	4.2189E-12

TABLE 6.5 (continued)

Matrix	Order	Gaussian elimination with minimum storage	Gaussian elimination	Actual
		Predicted		
A_4	17	1.3331E-13	2.6149E-11	2.3157E-14
A_5	17	4.3193E-11	2.3488E-9	5.3347E-12
A_4	18	1.4766E-13	3.4181E-11	2.7709E-14
A_5	18	5.3643E-11	3.4426E-9	4.6286E-11
A_4	19	1.6361E-13	4.4308E-11	3.3369E-14
A_5	19	6.6299E-11	4.9776E-9	2.6218E-11
A_4	20	1.7952E-13	5.6434E-11	3.9568E-14
A_5	20	8.0606E-11	7.0247E-9	5.4079E-11

TABLE 6.6

BAND MATRICES-RELATIVE ERROR

IN SOLUTION OF EQUATIONS

Coefficient Matrix	Order	Right-hand Side	Gaussian elimination with minimum storage	Gaussian elimination	Actual
			Predicted		
A_4	5	Vector of 2s'	7.4607E-15	4.6663E-14	2.9606E-16
A_5	5	"	4.8697E-14	8.4437E-14	6.3845E-15
A_4	10	"	6.8389E-15	2.9322E-13	5.9212E-16
A_5	10	"	5.9640E-14	7.0889E-13	9.7681E-15
A_4	11	"	6.7824E-15	3.8117E-13	3.9475E-16
A_5	11	"	6.0431E-14	9.4153E-13	1.5137E-14
A_4	12	"	6.7353E-15	4.8512E-13	5.0753E-16
A_5	12	"	6.1072E-14	1.2194E-12	2.0086E-14
A_4	13	"	6.6954E-15	6.0638E-13	4.3502E-16
A_5	13	"	6.1602E-14	1.5466E-12	2.4163E-14

TABLE 6.6 (continued)

Coefficient Matrix	Order	Right-hand Side	Gaussian elimination with minimum storage	Gaussian elimination	Actual
			Predicted		
A_4	14	"	6.6613E-15	7.4630E-13	4.4409E-16
A_5	14	"	6.2048E-14	1.9272E-12	3.0512E-14
A_4	15	"	6.6317E-15	9.0621E-13	0.0
A_5	15	"	6.2429E-14	2.3650E-12	3.7628E-14
A_4	16	"	6.6058E-15	1.0874E-12	3.9474E-16
A_5	16	"	6.2757E-14	2.8640E-12	3.8102E-14
A_4	17	"	6.5829E-15	1.2913E-12	3.5088E-16
A_5	17	"	6.3044E-14	3.4284E-12	3.8399E-14
A_4	18	"	6.5626E-15	1.5192E-12	3.5527E-16
A_5	18	"	6.3296E-14	5.0620E-11	2.6398E-14
A_4	19	"	6.5440E-15	1.7723E-12	2.8421E-16
A_5	19	"	6.3520E-14	4.7689E-12	1.3886E-14
A_4	20	"	6.5281E-15	2.0521E-12	3.2297E-16
A_5	20	"	6.7320E-14	5.5531E-12	2.1632E-14

CHAPTER VII

CONCLUSIONS AND SUGGESTIONS FOR FUTURE RESEARCH

7.1 Conclusions

The following conclusions are drawn from the numerical results:

1. As can be expected, the predicted round-off error incurred in inverting matrices and solving systems of equations using the algorithms described in Chapter III are substantially larger than the actual round-off error. This is due to the fact that the theoretical results are obtained after allowing for the maximum round-off error in each calculation and no allowance is made for the statistical effect. The results obtained in Chapter III therefore represent upper bounds which are, in general, much higher than the actual errors. It is important, however, to note that certain matrices can be constructed such that the actual round-off error is close to the predicted bound.

2. The upper bounds for the round-off error incurred in computing the inverse and solution of a system of equations using Schur's algorithms, described in Chapter III, are comparatively larger than the corresponding errors using

Gaussian elimination. This supports the remark made in Section 3.8 that the dominant term in the expression for round-off error using Schur's algorithms is of higher order than the corresponding term for Gaussian elimination.

3. The predicted and actual values of the round-off errors incurred in computing the inverse of a block-symmetric matrix and solution of a system of equations using Charmonman's algorithms are smaller compared to those obtained by using Schur's algorithms and Gaussian elimination, in most cases. This suggests that Charmonman's algorithms are less susceptible to the accumulation of round-off. These remarks are in complete agreement with the observations made in Sections 4.5 and 4.8 that the dominant term in the expression for round-off error using Schur's algorithms is of higher order than the corresponding term for Charmonman's algorithms.

4. The predicted and actual round-off errors in inverting band matrices and solving systems of equations with band matrices are almost equal, whereas the predicted round-off error using Gaussian elimination without minimum storage is comparatively larger than the actual value. This is due to the fact tnat only a few arithmetic operations

are involved in inverting band matrices and solving systems of equations with band matrices, and that all computations using APL\360 are done in double-precision.

7.2 Suggestions for Future Research

1. The error bounds obtained in Chapters III, IV and V could be significantly improved if the analysis is carried out using accumulation of inner products in floating-point.

2. Winograd [20], has proposed a new algorithm for matrix multiplication which requires smaller number of multiplications. This algorithm can be applied to invert matrices and solve systems of equations using methods described in Chapters III, IV and V, provided partitioned matrices and/or whole matrices are inverted using a variation of Gaussian elimination given in [3, pp. 136-137] or [20]. Perhaps, better error bounds for the predicted round-off error incurred in the computation of inverse of a matrix and the solution of a system of equations, could be obtained using the new algorithm for inner products.

BIBLIOGRAPHY

1. Charmonman, S., "An Efficient Algorithm for Inverting a Block-Symmetric Matrix", Math. Comp., 21, 1967, pp. 715-717.

2. Charmonman, S. and Julius, R.S., "Explicit Inverses and Condition Numbers of Certain Circulants", Math. Comp., 22, 1968, pp. 428-430.

3. Faddeev, D.K. and Faddeeva, V.N., Computational Methods of Linear Algebra (Translated by Robert C. Williams), W.H. Freeman and Company, San Francisco, 1963.

4. Forsythe, G.E. and Moler, C.B., Computer Solution of Linear Algebraic Systems, Prentice-Hall, Inc., Englewood Cliffs, N.J., 1967.

5. Fox, L., An Introduction to Numerical Linear Algebra, Clarendon Press, Oxford, 1964.

6. Householder, A.S., "The Approximate Solution of Matrix Problems", J. Assoc. Comp. Mach., 5, 1958, pp. 205-243.

7. International Business Machines, IBM System/360: Scientific Subroutine Package 360-CM-03X, Version II, Programmers Manual, White Plains, N.Y.,1966.

8. Korganoff, A. and Pavel-Parvu, M., Éléments de Théorie des Matrices Carrées et Rectangles en Analyse Numérique, Dunod, Paris, 1967.

9. Marcus, M., "Basic Theorems in Matrix Theory", Nat. Bur. Standards Appl. Math. Ser., 57, 1960, p. 17.

10. Newman, M. and Todd, J., "The Evaluation of Matrix Inversion Programs", J. Soc. Indust. Appl. Math., 6, 1958, pp. 466-476.

11. Ralston, A., A First Course in Numerical Analysis, McGraw-Hill Book Company, New York, 1965.

12. Turing, A.M., "Rounding-off errors in Matrix Processes", Quart. J. Mech., 1, 1948, pp. 287-308.

13. Von Neumann, J. and Goldstine, H.H., "Numerical Inverting of Matrices of High Order", Bull. Amer. Math. Soc., 53, 1947, pp. 1021-1099.

14. Wilkinson, J.H., "Error Analysis of Floating-Point Computation", Numer. Math., 2, 1960, pp. 319-340.

15. Wilkinson, J.H., "Rounding Errors in Algebraic Processes", Information Processing: UNESCO; Munich: Oldenburg; London: Buttersworth, 1960, pp. 44-53.

16. Wilkinson, J.H., "Error Analysis of Direct Methods of Matrix Inversion", J. Assoc. Comp. Mach., 8, 1961, pp. 281-330.

17. Wilkinson, J.H., Rounding Errors in Algebraic Processes, Prentice-Hall, Inc., Englewood Cliffs, N.J., 1963.

18. Wilkinson, J.H., The Algebraic Eigenvalue Problem, Clarendon Press, Oxford, 1965.

19. Wilkinson, J.H., "The Solution of Ill-Conditioned Linear Equations", Chapter III of Mathematical Methods for Digital Computers, Vol. 2, edited by A. Ralston and H. Wilf, Wiley and Sons, Inc., New York, 1967.

20. Winograd, S., "A New Algorithm for Inner Product", IBM Watson Research Center, York Town Heights, N.Y. Research Report Rc-1947, November 21, 1967.

APPENDIX

LISTINGS OF FORTRAN IV AND APL\360 PROGRAMS

Listings of FORTRAN IV subprograms and APL\360 functions for inverting matrices and solving systems of linear algebraic equations are included in the following pages. The SUBROUTINE GAUSSC computes the inverse of a matrix or solution of a system of linear equations using Gaussian elimination with complete pivoting. The SUBROUTINE ERRBD2, SUBROUTINE ERRBD4 and SUBROUTINE ERRBD5 compute the inverse of a general and block-symmetric matrix. The SUBROUTINE ERRBD3 and ERRBD6 compute the solution of system of equations with general and block-symmetric coefficient matrices. The SUBROUTINE SCHURI constructs the inverse of a partitioned matrix. The SUBROUTINE MADD and SUBROUTINE MPROD compute the sum and product of two matrices, respectively. The SUBROUTINE NORM computes the ∞-norm of an arbitrary matrix and/or the element of maximum modulus of a vector.

The APL function BAND computes the inverse and solution of equations using minimum storage for band matrices.

The unit round-off error for IBM System 360/67 computer is 2^{-21} for single-precision arithmetic and 2^{-53}

for double-precision arithmetic. Although the word length in single and double-precision arithmetic is 32 and 64 bits, 1 bit is used for the sign, 7 bits for the exponent, and three leading bits of the fractional part may be zero, since the normalization is done in hexadecimal.

```
      SUBROUTINE GAUSSC(N,M,A,X,SOL,G)
      DIMENSION A(20,20),X(20,20),SOL(20,20),NC(20),NR(20)
C
C THIS SUBROUTINE COMPUTES THE INVERSE OF A MATRIX OR
C SOLUTION OF A SYSTEM OF EQUATIONS OF ORDER N BY
C GAUSSIAN ELIMINATION WITH COMPLETE PIVOTING.THE INVERSE
C OR SOLUTION VECTOR IS RETURNED VIA THE ARRAY SOL.
C M IS THE NUMBER OF RIGHT-HAND SIDES.
C X IS THE MATRIX OF RIGHT-HAND SIDES.
C G IS THE ELEMENT OF MAXIMUM MAGNITUDE AT ALL STAGES IN
C THE REDUCTION OF A INTO UPPER-TRIANGULAR MATRIX.IT IS
C USED IN THE COMPUTATION OF THE PREDICTED RELATIVE ERROR
C FOR VARIOUS ALGORITHMS.
C
      COMMON FLAG
      REAL MIJ
      IF(N.EQ.1)GO TO 35
C
C DECLARE ZERO CRITERION
C
      ZERO=1E-8
C
C CHECK THE NUMBER OF EQUATIONS GIVEN.
C
      IF(M.EQ.1)GO TO 4
      DO 2 I=1,N
      DO 2 J=1,N
 2    X(I,J)=0
      DO 3 I=1,N
 3    X(I,I)=1
 4    NM1=N-1
      NP1=N+1
C
C THE BEGINNING OF FORWARD ELIMINATION.
C
      DO 31 I=1,NM1
      IP1=I+1
      NR(I)=I
      NC(I)=I
C
C USE OF COMPLETE PIVOTING.
C
      PIVOT=0
      DO 5 J=I,N
      DO 5 K=I,N
      IF(PIVOT.GE.ABS(A(J,K)))GO TO 5
      PIVOT=ABS(A(J,K))
C
C THE VECTORS NR AND NC NOTE THE NECESSARY ROW AND COLUMN
C INTERCHANGES,RESPECTIVELY.
C
```

```
      NR(I)=J
      NC(I)=K
 5    CONTINUE
C
C IF,AT ANY STAGE OF REDUCTION OF A ,A PIVOTAL ELEMENT IS
C LESS THAN THE ZERO CRITERION ,THE MATRIX IS FOUND TO BE
C SINGULAR.
C
      IF(PIVOT.LE.ZERO)GO TO 65
C
C INTERCHANGE ROWS,IF NECESSARY.
C
      IF(NR(I).EQ.I)GO TO 15
      DO 10 J=1,N
      TEMP=A(I,J)
      A(I,J)=A(NR(I),J)
      A(NR(I),J)=TEMP
 10   CONTINUE
      DO 11 J=1,M
      TEMP=X(I,J)
      X(I,J)=X(NR(I),J)
      X(NR(I),J)=TEMP
 11   CONTINUE
C
C INTERCHANGE COLUMNS ,IF NECESSARY.
C
 15   IF(NC(I).EQ.I)GO TO 25
      DO 20 J=1,N
      TEMP=A(J,I)
      A(J,I)=A(J,NC(I))
      A(J,NC(I))=TEMP
 20   CONTINUE
 25   DO 31 J=IP1,N
      MIJ=-A(J,I)/A(I,I)
      DO 30 K=IP1,N
      A(J,K)=A(J,K)+MIJ*A(I,K)
 30   CONTINUE
      DO 31 L=1,M
      X(J,L)=X(J,L)+MIJ*X(I,L)
 31   CONTINUE
C
C IF,THE A(N,N) ELEMENT OF A AFTER N-1 REDUCTIONS IS LESS
C THAN ZERO CRITERION,THE MATRIX A IS FOUND TO BE
C SINGULAR.
C
 35   IF(ABS(A(N,N)).LE.ZERO)GO TO 70
C
C THE BEGINNING OF BACK-SUBSTITUTION.
C
      DO 40 I=1,M
      X(N,I)=X(N,I)/A(N,N)
```

```
 40   CONTINUE
      IF(N.EQ.1)GO TO 75
      DO 50 K=2,N
      I=NP1-K
      IP1=I+1
      DO 50 L=1,M
      SUM=0
      DO 45 J=IP1,N
      SUM=SUM+A(I,J)*X(J,L)
 45   CONTINUE
      X(I,L)=(X(I,L)-SUM)/A(I,I)
 50   CONTINUE
C
C BACK-INTERCHANGE OF COLUMNS.
C
      DO 60 K=1,NM1
      I=N-K
      IF(NC(I).EQ.I)GO TO 60
      DO 55 J=1,M
      TEMP=X(I,J)
      X(I,J)=X(NC(I),J)
      X(NC(I),J)=TEMP
 55   CONTINUE
 60   CONTINUE
C
C STORE INVERSE OR SOLUTION VECTOR IN ARRAY SOL.
C
      DO 61 I=1,N
      DO 61 J=1,M
 61   SOL(I,J)=X(I,J)
C
C COMPUTE ELEMENT OF MAXIMUM MODULUS IN THE REDUCTION OF A.
C
      G=0
      DO 62 I=1,N
      DO 62 J=I,N
 62   G=AMAX1(G,ABS(A(I,J)))
      GO TO 75
 65   FLAG=1
      GO TO 75
 70   FLAG=2
 75   RETURN
      END
```

```
      SUBROUTINE ERRBD1(N,NRA,A,B,C,D,RENORM,TNORM,HC,
     1PROD,EN2)
      DIMENSION A(20,20),B(20,20),C(20,20),D(20,20),
     1HC(20,20),X(20,20),ACI(20,20),PROD(20,20),
     2PROD1(20,20),DELTA(20,20)
C
C THIS SUBROUTINE COMPUTES THE MATRIX HC USING SUBMATRICES
C A,B,C AND D AND COMPUTES AN UPPER BOUND FOR THE
C ABSOLUTE ROUND-OFF ERROR EN2 INCURRED IN THE
C CALCULATION OF HC.THE COMPUTED HC AND EN2 ARE RETURNED
C VIA HC AND EN2.THE PRODUCT ACI*B,WHERE ACI IS THE
C COMPUTED INVERSE OF A IS ALSO RETURNED VIA ARRAY PROD
C FOR FUTURE USE.
C N IS THE ORDER OF THE MATRIX R.
C NRA IS THE ORDER OF THE MATRIX A.
C RENORM IS THE INFINITY NORM OF THE EXACT INVERSE OF R.
C TNORM IS THE INFINITY NORM OF THE EXACT INVERSE OF A.
C EP IS THE UNIT ROUND-OFF ERROR FOR IBM 360/67 COMPUTER.
C
      DOUBLE PRECISION EP
      EP=2.D0**(-21)
      NMR=N-NRA
C
C BNORM,CNORM AND DNORM IS THE INFINITY NORM OF MATRICES
C B,C AND D,RESPECTIVELY.
C
      CALL NORM(NRA,NMR,B,BNORM)
      CALL NORM(NMR,NRA,C,CNORM)
      CALL NORM(NMR,NMR,D,DNORM)
C
C COMPUTE ACI,THE INVERSE OF A.
C
      CALL GAUSSC(NRA,NRA,A,X,ACI,G)
C
C ACNORM IS THE INFINITY NORM OF ACI.
C
      CALL NORM(NRA,NRA,ACI,ACNORM)
      EN11=2.005*NRA**2+NRA**3
      EN12=EN11*TNORM*G
      EN13=EN12+1.+2.*NRA+NRA*(NRA+2.)*EP
      EN14=EN13*CNORM*ACNORM
      EN1=EP*(DNORM+EN14*BNORM)
C
C COMPUTE PROD,HC AND EN2.
C
      CALL MPROD(NRA,NRA,NMR,ACI,B,PROD)
      CALL MPROD(NMR,NRA,NMR,C,PROD,PROD1)
      CALL MADD(NMR,NMR,D,PROD1,DELTA)
      CALL GAUSSC(NMR,NMR,DELTA,X,HC,G)
      CALL NORM(NMR,NMR,HC,HCNORM)
      EN21=2.005*NMR**2+NMR**3
```

```
      EN22=EP*EN21*G*HCNORM
      EN23=RENORM*EN1
      EN2=(EN22+EN23)*(RENORM/(1.-RENORM*EN1))
      RETURN
      END
```

```
      SUBROUTINE ERRBD2(N,NRA,A,B,C,D,RENORM,TNORM,RCI,
     1ERRS)
      DIMENSION A(20,20),B(20,20),C(20,20),D(20,20),
     1RCI(20,20),PROD(20,20),PROD1(20,20),EC(20,20),
     2FC(20,20),GC(20,20),HC(20,20),ACI(20,20),X(20,20),
     3S(20,20),DC(20,20)
C
C THIS SUBROUTINE COMPUTES THE INVERSE OF R USING SCHUR'S
C ALGORITHM AND CALCULATES UPPER BOUND FOR THE ROUND-OFF
C ERROR INCURRED IN THE CALCULATION.THE COMPUTED INVERSE
C IS RETURNED VIA THE ARRAY RCI.THE PREDICTED RELATIVE
C ERROR IS STORED IN ERRS.
C N IS THE ORDER OF THE MATRIX R.
C NRA IS THE ORDER OF MATRIX A.
C A,B,C AND D ARE SUBMATRICES OF R.
C RENORM IS THE INFINITY NORM OF THE EXACT INVERSE OF R.
C TNORM IS THE INFINITY NORM OF THE EXACT INVERSE OF A.
C EP IS THE UNIT ROUND-OFF ERROR FOR IBM 360/67 COMPUTER.
C
      DOUBLE PRECISION EP
      EP=2.D0**(-21)
      NMR=N-NRA
      NRAP1=NRA+1
      DO 5 I=1,NRA
      DO 5 J=1,NRA
 5    S(I,J)=A(I,J)
C
C STORE A FOR FUTURE USE.
C
C
C COMPUTE HC.
C
      CALL ERRBD1(N,NRA,A,B,C,D,RENORM,TNORM,HC,PROD,EN2)
C
C G IS THE ELEMENT OF MAXIMUM MODULUS IN THE REDUCTION
C OF A.
C
      CALL GAUSSC(NRA,NRA,S,X,ACI,G)
      CALL NORM(NRA,NRA,ACI,ACNORM)
      CALL NORM(NRA,NMR,B,BNORM)
      CALL NORM(NMR,NRA,C,CNORM)
C
C HCNORM IS THE INFINITY NORM OF THE MATRIX HC.
C
      CALL NORM(NMR,NMR,HC,HCNORM)
      EN11=2.005*NRA**2+NRA**3
      EN12=EN11*TNORM*G
      EN31=TNORM*BNORM*EN2
      EN32=(EN12+N+NRA*NMR)*EP
      EN3=EN31+EN32*ACNORM*BNORM*HCNORM
      EN41=EN2+EN12*RENORM+EP*(N+NRA*NMR*EP)*HCNORM
```

```
      EN4=EN41*ACNORM*CNORM
C
C COMPUTE GC.
C
      CALL MPROD(NMR,NMR,NRA,HC,C,PROD1)
      CALL MPROD(NMR,NRA,NRA,PROD1,ACI,GC)
      DO 10 I=1,NMR
      DO 10 J=1,NRA
 10   GC(I,J)=-1.*GC(I,J)
C
C GCNORM IS THE INFINITY NORM OF THE MATRIX GC.
C
      CALL NORM(NMR,NRA,GC,GCNORM)
      EN51=TNORM*BNORM*EN4
      EN52=(1.+EN12)*ACNORM
      EN53=N*(1.+EP)+NRA*NMR*EP
      EN54=(EN52+EN53*ACNORM)*BNORM*GCNORM
      EN5=EP*(EN51+EN52+EN54)
C
C COMPUTE RELATIVE ERROR ERRS.
C
      ERR1=EN2+EN4
      ERR2=EN3+EN5
      ERRS=AMAX1(ERR1,ERR2)
      ERRS=ERRS/RENORM
C
C COMPUTE FC
C
      CALL MPROD(NRA,NMR,NMR,PROD,HC,FC)
      DO 20 I=1,NRA
      DO 20 J=1,NMR
 20   FC(I,J)=-1.*FC(I,J)
C
C COMPUTE EC.
C
      CALL MPROD(NRA,NMR,NRA,PROD,GC,PROD1)
      CALL MADD(NRA,NRA,ACI,PROD1,EC)
      DO 103 I=1,NMR
      DO 103 J=1,NMR
      DC(I,J)=HC(I,J)
 103  CONTINUE
C
C COMPUTE RCI.
C
      CALL SCHURI(N,NRA,EC,FC,GC,DC,RCI)
      RETURN
      END
```

```
      SUBROUTINE ERRBD3(N,NRA,A,B,C,D,RHS,TNORM,RENORM,
     1XEMAX,XC1,ERRS)
      DIMENSION A(20,20),B(20,20),C(20,20),D(20,20),
     1RHS(20,20),XC2(20,20),X(20,20),ACI(20,20),B1(20,20),
     2B2(20,20),PROD(20,20),PROD1(20,20),PROD2(20,20),
     3DELTA(20,20),DEL(20,20),XC1(20,20),DCI(20,20),
     4RS(20,20),S(20,20)
C
C THIS SUBROUTINE COMPUTES THE SOLUTION OF A SYSTEM OF
C EQUATIONS USING SCHUR'S ALGORITHM AND CALCULATES AN
C UPPER BOUND FOR THE ROUND-OFF ERROR INCURRED IN THE
C CALCULATION .THE COMPUTED SOLUTION IS RETURNED VIA ARRAY
C XC1 AND THE PREDICTED RELATIVE ERROR IS STORED IN ERRS.
C N IS THE ORDER OF THE COEFFICIENT MATRIX R.
C NRA IS THE ORDER OF THE MATRIX A.
C A,B,C AND D ARE THE SUBMATRICES OF R.
C RHS IS THE MATRIX OF RIGHT-HAND SIDES.
C TNORM IS THE INFINITY NORM OF THE EXACT INVERSE OF A.
C RENORM IS THE INFINITY NORM OF THE EXACT INVERSE OF R.
C XEMAX IS THE INFINITY NORM OF THE EXACT SOLUTION OF THE
C GIVEN SYSTEM OF EQUATIONS.
C EP IS THE UNIT ROUND-OFF ERROR FOR THE IBM 360/67
C COMPUTER.
C
      DOUBLE PRECISION EP
      EP=2.D0**(-21)
      NMR=N-NRA
C
C STORE THE MATRIX A FOR FUTURE USE.
C
      DO 5 I=1,NRA
      DO 5 J=1,NRA
      S(I,J)=A(I,J)
 5    CONTINUE
C
C ANORM IS THE INFINITY NORM OF THE MATRIX A.
C
      CALL NORM(NRA,NRA,A,ANORM)
C
C PARTITION THE RIGHT-HAND SIDE RHS.
C
      DO 10 I=1,NRA
 10   B1(I,1)=RHS(I,1)
      NRAP1=NRA+1
      DO 20 I=NRAP1,N
      I1=I-NRA
 20   B2(I1,1)=RHS(I,1)
C
C ACI IS THE COMPUTED INVERSE OF A.
C
      CALL GAUSSC(NRA,NRA,A,X,ACI,G)
```

```
      CALL NORM(NRA,1,B1,AMAXB1)
      CALL NORM(NMR,1,B2,AMAXB2)
      EN11=2.005*NRA**2+NRA**3
      EN12=EN11*TNORM*G
      EN13=EN12+1.+2.*NRA+NRA*(NRA+2.)*EP
C
C CNORM IS THE INFINITY NORM OF THE MATRIX C.
C
      CALL NORM(NMR,NRA,C,CNORM)
      CALL NORM(NRA,NRA,ACI,ACNORM)
      EN14=EN13*CNORM*ACNORM
      ENS2=EP*(AMAXB2+EN14*AMAXB1)
C
C COMPUTE THE COEFFICIENT MATRIX AND THE RIGHT-HAND SIDE
C FOR COMPUTING THE SOLUTION VECTOR OF ORDER N-NRA.
C
      CALL MPROD(NMR,NRA,NRA,C,ACI,PROD)
      CALL MPROD(NMR,NRA,1,PROD,B1,PROD1)
      CALL MADD(NMR,1,B2,PROD1,X)
      CALL MPROD(NMR,NRA,NMR,PROD,B,PROD1)
      CALL MADD(NMR,NMR,D,PROD1,DELTA)
      CALL GAUSSC(NMR,1,DELTA,X,XC2,G)
      CALL NORM(NMR,1,XC2,AMAXC2)
      EK1=EP*EN11*G
C
C DNORM IS THE INFINITY NORM OF MATRIX D.
C
      CALL NORM(NMR,NMR,D,DNORM)
      EN1=EP*(DNORM+EN14*BNORM)
      EN3=(ENS2+(EN1+EK1)*AMAXC2)*RENORM
      IF(NRA-NMR)15,21,15
C
C COMPUTE THE SOLUTION VECTOR OF ORDER NRA,IF NRA NOT
C EQUAL TO N/2.
C
 15   EN41=1.+2.*NMR+NMR*(NMR+2.)*EP
      DENORM=RENORM
      CALL GAUSSC(NMR,NMR,D,X,DCI,G)
      CALL NORM(NMR,NMR,DCI,DCNORM)
      EN42=2.005*NMR**2+NMR**3
      EN43=(G*EN42*DENORM+EN41)*BNORM*DCNORM
      EN4=EP*(ANORM+EN43*CNORM)
      EN5=EP*(AMAXB1+EN43*AMAXB2)
      CALL MPROD(NRA,NMR,NMR,B,DCI,PROD)
      CALL MPROD(NRA,NMR,NRA,PROD,C,PROD1)
      CALL MPROD(NRA,NMR,1,PROD,B2,PROD2)
      CALL MADD(NRA,NRA,S,PROD1,DEL)
      CALL MADD(NRA,1,B1,PROD2,RS)
C
C COMPUTE THE SOLUTION VECTOR OF ORDER NRA.
C
```

```
      CALL GAUSSC(NRA,1,DEL,RS,XC1,G)
      CALL NORM(NRA,1,XC1,AMAXC1)
      EK2=EP*EN11*G
      EN6=(EN5+(EN4+EK2)*AMAXC1)*RENORM
      GO TO 25
C
C COMPUTE THE SOLUTION VECTOR OF ORDER N/2,IF NRA EQUAL
C TO N/2.
C
 21   CENORM=RENORM
      CALL MPROD(NMR,NMR,1,D,XC2,PROD)
      CALL MADD(NMR,1,B2,PROD,RS)
      CALL GAUSSC(NMR,1,C,RS,XC1,G)
      CALL NORM(NMR,1,XC1,AMAXC1)
      EN6=(EN4+EK2*AMAXC1)*CENORM
 25   ERRS=AMAX1(EN3,EN6)
C
C COMPUTE THE PREDICTED RELATIVE ERROR ERRS.
C
      ERRS=ERRS/XEMAX
C
C CONSTRUCT SOLUTION VECTOR OF ORDER N.
C
      DO 30 I=NRAP1,N
 30   XC1(I,1)=XC2(I-NRA,1)
      RETURN
      END
```

```
      SUBROUTINE ERRBD4(N,A,B,C,D,RENORM,RCI,ERRS)
      DIMENSION A(20,20),B(20,20),RCI(20,20),EC(20,20),
     1FC(20,20),PROD(20,20),S(20,20),C(20,20),D(20,20),
     2HC(20,20)
C
C THIS SUBROUTINE COMPUTES THE INVERSE OF A BLOCK-
C SYMMETRIC MATRIX USING SCHUR'S ALGORITHM AND CALCULATES
C AN UPPER BOUND FOR THE ROUND-OFF ERROR INCURRED IN THE
C CALCULATION.THE INVERSE OF THE BLOCK-SYMMETRIC MATRIX
C IS RETURNED VIA THE ARRAY RCI AND THE PREDICTED RELATIVE
C ERROR IS STORED IN ERRS.
C N IS THE ORDER OF THE MATRIX R.
C A,B,C AND D ARE SUBMATRICES OF R.
C RENORM IS THE INFINITY NORM OF THE EXACT INVERSE OF R.
C EP IS THE UNIT ROUND-OFF ERROR FOR THE IBM 360/67
C COMPUTER.
C
      DOUBLE PRECISION EP
      EP=2.D0**(-21)
      NRA=N/2
      NMR=N/2
      NRAP1=NRA+1
C
C STORE MATRIX A FOR FUTURE USE.
C
      DO 5 I=1,NRA
      DO 5 J=1,NRA
      S(I,J)=A(I,J)
 5    CONTINUE
      TNORM=RENORM
C
C COMPUTE MATRIX EC.
C
      CALL ERRBD1(N,NRA,A,B,C,D,RENORM,TNORM,HC,PROD,EN2)
      DO 10 I=1,NMR
      DO 10 J=1,NMR
 10   EC(I,J)=HC(I,J)
      E2=EN2
C
C COMPUTE MATRIX FC.
C
      CALL ERRBD1(N,NRA,D,C,B,S,RENORM,TNORM,HC,PROD,EN2)
      E4=EN2
      DO 20 I=1,NMR
      DO 20 J=1,NMR
 20   FC(I,J)=HC(I,J)
      ERRS=E2+E4
C
C COMPUTE RELATIVE ERROR ERRS.
C
      ERRS=ERRS/RENORM
```

```
C
C CONSTRUCT INVERSE RCI USING EC AND FC.
C
      CALL SCHURI(N,NRA,EC,FC,FC,EC,RCI)
      RETURN
      END
```

```
      SUBROUTINE ERRBD5(N,A,B,RENORM,RC,ERRC)
      DIMENSION A(20,20),B(20,20),RC(20,20),P(20,20),
     1Q(20,20),PCI(20,20),QCI(20,20),X(20,20),EC(20,20),
     2FC(20,20)
C
C THIS SUBROUTINE COMPUTES THE INVERSE OF A BLOCK-
C SYMMETRIC MATRIX USING CHARMUNMAN'S ALGORITHM AND
C CALCULATES AN UPPER BOUND FOR THE ROUND-OFF ERROR
C INCURRED IN THE CALCULATION.THE INVERSE IS RETURNED VIA
C THE ARRAY RC AND THE PREDICTED RELATIVE ERROR IS STORED
C IN ERRC.
C N IS THE ORDER OF THE BLOCK-SYMMETRIC MATRIX.
C A AND B ARE THE SUBMATRICES OF R OF ORDER N/2 EACH.
C RENORM IS THE INFINITY NORM OF THE EXACT INVERSE OF R.
C EP IS THE UNIT ROUND-OFF ERROR FOR THE IBM 360/67
C COMPUTER.
C
      DOUBLE PRECISION EP
      EP=2.D0**(-21)
      NRA=N/2
      CALL NORM(NRA,NRA,A,ANORM)
      CALL NORM(NRA,NRA,B,BNORM)
      EN1=EP*(ANORM+BNORM)
      EN2=2.*EP*(ANORM+2.*BNORM)
      EN31=2.005*NRA**2+NRA**3
C
C COMPUTE P AND Q.
C
      DO 10 I=1,NRA
      DO 10 J=1,NRA
      P(I,J)=A(I,J)+B(I,J)
 10   Q(I,J)=P(I,J)-2.*B(I,J)
C
C PCI IS THE COMPUTED INVERSE OF P.
C
      CALL GAUSSC(NRA,NRA,P,X,PCI,G)
      CALL NORM(NRA,NRA,PCI,PCNORM)
      PENORM=2.*RENORM
      QENORM=2.*RENORM
      EN32=G*EN31*PCNORM+PENORM*EN1
      EN3=EP*EN32*(PENORM/(1.-PENORM*EN1))
C
C QCI IS THE COMPUTED INVERSE OF Q.
C
      CALL GAUSSC(NRA,NRA,Q,X,QCI,G)
      CALL NORM(NRA,NRA,QCI,QCNORM)
      EN41=G*EN31*QCNORM+QENORM*EN2
      EN4=EP*EN41*(QENORM/(1.-QENORM*EN2))
      EN5=0.5*(EN3+EN4)+0.5*EP*(PCNORM+QCNORM)
      ERRC=EN4+2.*EN5
C
```

```
C COMPUTE THE PREDICTED RELATIVE ERROR ERRC.
C
      ERRC=ERRC/RENORM
C
C COMPUTE EC.
C
      DO 20 I=1,NRA
      DO 20 J=1,NRA
 20   EC(I,J)=0.5*(PCI(I,J)+QCI(I,J))
C
C COMPUTE FC.
C
      CALL MADD(NRA,NRA,EC,QCI,FC)
C
C CONSTRUCT INVERSE OF R USING EC AND FC.
C
      CALL SCHURI(N,NRA,EC,FC,FC,EC,RC)
      RETURN
      END
```

```
      SUBROUTINE ERRBD6(N,A,B,RHS,RENORM,XEMAX,XC2,ERRC)
      DIMENSION A(20,20),B(20,20),RHS(20,20),YC2(20,20),
     1B1(20,20),B2(20,20),RS1(20,20),RS2(20,20),P(20,20),
     2Q(20,20),YC1(20,20),XC1(20,20),XC2(20,20),SOL(20,20)
C
C THIS SUBROUTINE COMPUTES THE SOLUTION OF A SYSTEM OF
C EQUATIONS WITH BLOCK-SYMMETRIC COEFFICIENT MATRIX USING
C CHARMONMAN'S ALGORITHM AND CALCULATES AN UPPER BOUND
C FOR THE ROUND-OFF ERROR INCURRED IN THE CALCULATION.THE
C COMPUTED SOLUTION IS RETURNED VIA THE ARRAY XC2 AND THE
C PREDICTED RELATIVE ERROR  IS STORED IN ERRC.
C N IS THE ORDER OF THE COEFFICIENT MATRIX R.
C A AND B ARE THE SUBMATRICES OF R.
C RHS IS THE MATRIX OF RIGHT-HAND SIDES.
C RENORM IS THE INFINITY NORM OF THE EXACT INVERSE OF R.
C XEMAX IS THE INFINITY NORM OF THE EXACT SOLUTION OF THE
C GIVEN SYSTEM OF EQUATIONS.
C EP IS THE UNIT ROUND-OFF ERROR FOR THE IBM 360/67
C COMPUTER.
C
      DOUBLE PRECISION EP
      EP=2.D0**(-21)
      NRA=N/2
      NRAP1=NRA+1
C
C PARTITION RIGHT-HAND SIDE RHS.
C
      DO 10 I=1,NRA
      B1(I,1)=RHS(I,1)
      I1=NRA+I
 10   B2(I,1)=RHS(I1,1)
      DO 20 I=1,NRA
      RS1(I,1)=B1(I,1)+B2(I,1)
 20   RS2(I,1)=RS1(I,1)-2.*B2(I,1)
C
C COMPUTE P AND Q.
C
      DO 30 I=1,NRA
      DO 30 J=1,NRA
      P(I,J)=A(I,J)+B(I,J)
 30   Q(I,J)=P(I,J)-2.*B(I,J)
      CALL NORM(NRA,NRA,A,ANORM)
      CALL NORM(NRA,NRA,B,BNORM)
      PENORM=2.*RENORM
      QENORM=2.*RENORM
      CALL NORM(NRA,1,B1,AMAXB1)
      CALL NORM(NRA,1,B2,AMAXB2)
C
C COMPUTE SOLUTION OF THE SYSTEM OF EQUATIONS WITH
C COEFFICIENT MATRIX P.
C
```

```
      CALL GAUSSC(NRA,1,P,RS1,YC1,G)
      CALL NORM(NRA,1,YC1,AMAXY1)
      EN51=(G*(2.005*NRA**2+NRA**3)+ANORM+BNORM)*AMAXY1
      EN52=AMAXB1+AMAXB2
      EN5=EP*(EN51+EN52)*PENORM
C
C COMPUTE SOLUTION OF THE SYSTEM OF EQUATIONS WITH
C COEFFICIENT MATRIX  Q.
C
      CALL GAUSSC(NRA,1,Q,RS2,YC2,G)
C
C COMPUTE FIRST N/2 COMPONENTS OF THE SOLUTION VECTOR.
C
      DO 35 I=1,NRA
 35   XC1(I,1)=0.5*(YC1(I,1)+YC2(I,1))
      CALL NORM(NRA,1,YC2,AMAXY2)
      EN61=(G*(2.005*NRA**2+NRA**3)+2.*(ANORM+2.*BNORM))
     1*AMAXY2
      EN6=EP*(EN53+EN61)*QENORM
C
C COMPUTE LAST N/2 COMPONENTS OF THE SOLUTION VECTOR.
C
      DO 40 I=1,NRA
 40   XC2(I,1)=XC1(I,1)-YC2(I,1)
      CALL NORM(NRA,1,XC1,AMAXD1)
      ERRC=EN5+2.*EN6+EP*AMAXD1
C
C COMPUTE THE PREDICTED RELATIVE ERROR ERRC.
C
      ERRC=ERRC/XEMAX
C
C CONSTRUCT SOLUTION VECTOR OF ORDER N.
C
      DO 50 I=NRAP1,N
 50   XC2(I,1)=XC1(I-NRA,1)
      RETURN
      END
```

```
      SUBROUTINE SCHURI(N,NRA,A,B,C,D,RCI)
      DIMENSION A(20,20),B(20,20),C(20,20),D(20,20),
     1RCI(20,20)
C
C THIS SUBROUTINE CONSTRUCTS A MATRIX OF ORDER N FROM
C THE SUBMATRICES A,B,C AND D,AND STORES IT IN ARRAY RCI.
C NRA IS THE ORDER OF MATRIX A.
C
      NRAP1=NRA+1
      DO 10 I=1,NRA
      DO 10 J=1,NRA
      RCI(I,J)=A(I,J)
 10   CONTINUE
      DO 15 I=1,NRA
      DO 15 J=NRAP1,N
      RCI(I,J)=B(I,J-NRA)
 15   CONTINUE
      DO 20 I=NRAP1,N
      DO 20 J=1,NRA
      RCI(I,J)=C(I-NRA,J)
 20   CONTINUE
      DO 25 I=NRAP1,N
      DO 25 J=NRAP1,N
      RCI(I,J)=D(I-NRA,J-NRA)
 25   CONTINUE
      RETURN
      END
```

```
      SUBROUTINE NORM(N,M,A,XNORM)
      DIMENSION A(20,20)
C
C THIS SUBROUTINE COMPUTES THE INFINITY NORM OF AN
C ARBITRARY MATRIX A OF DIMENSION N-BY-M AND STORES IT IN
C XNORM.
C
      XNORM=0
      DO 20 I=1,N
      SUM=0
      DO 10 J=1,M
 10   SUM=SUM+ABS(A(I,J))
      IF(XNORM.GT.SUM)GO TO 20
      XNORM=SUM
 20   CONTINUE
      RETURN
      END
```

```
      SUBROUTINE MPROD(N,M,K,A,B,AMLT)
      DIMENSION A(20,20),B(20,20),AMLT(20,20)
C
C THIS SUBROUTINE COMPUTES THE PRODUCT OF TWO MATRICES
C CONFORMABLE FOR MATRIX MULTIPLICATION.THE PRODUCT IS
C RETURNED VIA THE ARRAY AMLT.
C
      DO 20 I=1,N
      DO 20 J=1,K
      SUM=0
      DO 15 L=1,M
 15   SUM=SUM+A(I,L)*B(L,J)
 20   AMLT(I,J)=SUM
      RETURN
      END
```

```
      SUBROUTINE MADD(N,M,A,B,ADD)
      DIMENSION A(20,20),B(20,20),ADD(20,20)
C
C THIS SUBROUTINE SUBTRACTS MATRIX B OF DIMENSION N-BY-M
C FROM MATRIX A OF THE SAME DIMENSION.THE RESULT IS STORED
C IN ARRAY ADD.
C
      DO 10 I=1,N
      DO 10 J=1,M
      ADD(I,J)=A(I,J)-B(I,J)
 10   CONTINUE
      RETURN
      END
```

```
    ∇ BAND

[1]   →((0<(¯2×M)+1+MC←1+MUD+MLD)∨((MLD<0)∨MUD<0))/
      75
[2]   MC←⌊/M,MC
[3]   MZ←((MU+1)÷2)×MU←MC-(MUD+1)
[4]   MA←(M×MC)-((ML+1)÷2)×ML←MC-(MLD+1)
[5]   MR←M-ML
[6]   →((MLD=0)∧(MUD=0))/62
[7]   →(MLD=0)/47
[8]   →(MZ=0)/13
[9]   I←J←1
[10]  A←A[⍳MUD+J],((MC-MUD+I)⍴0),A[((⍴A)ω(⍴A)-MUD+J)/⍳⍴A]
[11]  J←J+MC+1
[12]  →(MU≥I←I+1)/10
[13]  PIV←⌈/|A
[14]  TOL←EPS×PIV
[15]  T←(MLD+1)⍴KL←KC←KI←KM←J←I←0
[16]  KM←I
[17]  →(I≤M-MUD+1)/19
[18]  KM←M-MUD
[19]  →(MR≥MLD+I)/25
[20]  KL←KL+1
[21]  →(KL<MUD)/25
[22]  →((⍴JK)≤2)/47
[23]  J←I+(⍳⍴JK)⍳KP←(|A[JK])⍳PIV←⌈/|A[JK←JK[1+⍳(⍴JK)-1]]
[24]  →26
[25]  J←I+(⍳⍴JK)⍳KP←(|A[JK])⍳PIV←⌈/|A[JK←(-(T←T+((MLD+1-KL)
      ⍴0),⍳KL))+1+MC×(¯1+⍳M)[KM+⍳MLD+1]]
[26]  →(PIV≤TOL)/77
[27]  →(J=I←I+1)/34
[28]  TEMP←A[JJ←JK[KP]+¯1+⍳MC-KI]
[29]  A[JJ]←A[S←JK[1]+¯1+⍳MC-KI]
[30]  A[S]←TEMP
[31]  TEMP←R[J;]
[32]  R[J;]←R[I;]
[33]  R[I;]←TEMP
[34]  K←2
[35]  R[I;]←R[I;]÷A[JK[1]]
[36]  A[S]←A[S←JK[1]+¯1+⍳MC-KI]÷A[JK[1]]
[37]  R[I+JK⍳JK[K-1];]←R[I+JK⍳JK[K-1];]-A[JK[K]]×R[I;]
[38]  A[LL]←((A[LL←JK[K]+¯1+⍳MC-KI]-A[JK[K]]×A[JK[1]+
      ¯1+⍳MC-KI])[1+⍳LC-KI+1]),0
[39]  →((⍴JK)≤2)/43
[40]  →((⍴JK)≤MLD)/42
[41]  →((MLD+1)≥K←K+1)/37
[42]  →(((⍴JK)+1)>K←K+1)/37
```

```
[43]  →(I≤N-NC)/46
[44]  KI←KI+1
[45]  →((((ρJK)=N+1-I)∧(I<M-1))/22
[46]  →(I<M-1)/16
[47]  R[M;]←R[M;]÷A[MA-MLD]
[48]  I←M-1
[49]  J←L←X←0
[50]  V←ι¯1+K←1
[51]  R[I;]←R[I;]-A[II←2+(K-1)++/((N,NC×(((M-ρN←MLD+ιNC-MLD
      +1)ρ1),0))[2+ιM-1])[J+ιN-1+J]]×R[M-L;]
[52]  V←V,II
[53]  →(L≤X)/56
[54]  K←K+1
[55]  →(X≤L←L-1)/51
[56]  R[I;]←R[I;]÷A[V[1]-1]
[57]  L←K+X
[58]  J←J+1
[59]  →((M-(MLD+MUD))≤I←I-1)/50
[60]  →(I≤0)/65
[61]  →50,X←X+1
[62]  I←1
[63]  R[I;]←R[I;]÷A[I]
[64]  →(M≥I←I+1)/63
[65]  C←⌈/|A
[66]  EP←2*¯53
[67]  F1←2×MLD×(3×MLD)+(3×MUD)+1
[68]  F2←(MLD+1)×(MLD+MUD+1)×((2×MLD)+MUD+4)
[69]  F3←0.5×EP×(F1+F2)×C
[70]  RCNORM←⌈/+/|R
[71]  →(TYPE=2)/73
[72]  →74,CREB←F3×RCNORM
[73]  CREB←(F3×RENORM×RCNORM)÷XFMAX
[74]  →0
[75]  'WRONG DATA'
[76]  →0
[77]  'THE COEFFICIENT MATRIX IS SINGULAR'
[78]  →0
      ∇
```